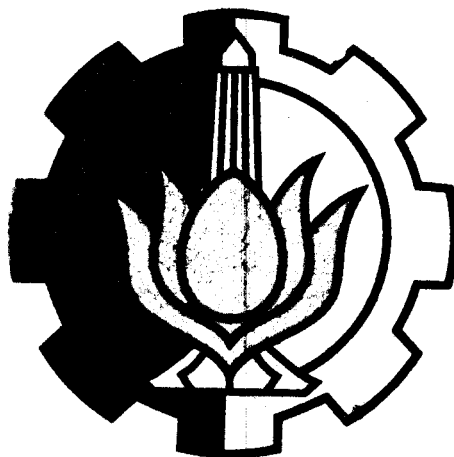


19.682/ITS/H/2004



MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH NOPEMBER

PENYELESAIAN MASALAH KNAPSACK DUA DIMENSI DENGAN ALGORITMA APROKSIMASI



REF
005.1
Hiz
P-1
1999

Disusun Oleh :

MUTIA AIZA

NRP. 2690.100.025

PERPUSTAKAAN ITS	
Tgl. Terima	16-7-2003
Terima Dari	H
No. Agenda Prp.	208378

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
1999**

**PENYELESAIAN MASALAH KNAPSACK
DUA DIMENSI DENGAN
ALGORITMA APROKSIMASI**

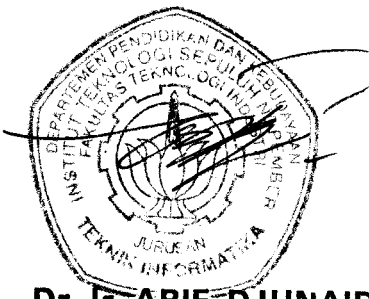
TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer
Pada**

**Jurusan Teknik Informatika
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
S u r a b a y a**

Mengetahui / Menyetujui

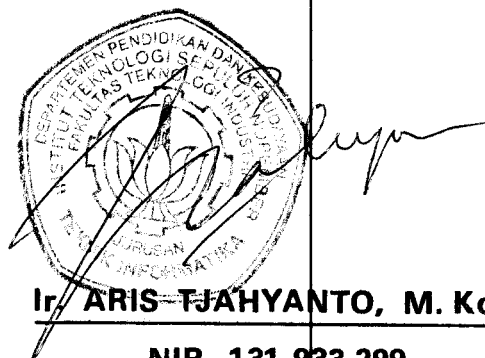
Dosen Pembimbing I



Dr. Ir. ARIF DJUNAIDY, M.Sc.

NIP. 131 633 403

Dosen Pembimbing II



Ir. ARIS TJAHYANTO, M. Kom

NIP. 131 933 299

**S U R A B A Y A
Pebruari, 1999**

Bismillahirrahmannirrahim,

Demi masa,

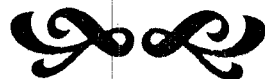
*Sesungguhnya manusia itu benar-benar berada dalam kerugian,
Kecuali orang-orang yang beriman dan mengerjakan amal saleh
dan nasehat menasihati supaya menaati kebenaran
dan nasehat menasihati supaya menetapi kesabaran.*

(Q.S. Al' Ashr 103:1-3)



kupersembahkan untuk:

Mamak & Bapak, yang tercinta
atas segala do'a, ketulusan dan kesabaran yang diberikan
Bang Is, kak Nova & de' Fadhil, yang tersayang
atas segala kasih sayang dan dukungan
serta Ghaxi & Jihan, yang menggugah



ABSTRAK

Efisiensi sangat diperlukan dalam setiap bidang, karena dengan efisiensi ini akan didapatkan suatu penghematan sumber daya. Di bidang permasalahan knapsack dua dimensi untuk pemotongan bahan, efisiensi penggunaan bahan sangat diperlukan untuk memaksimalkan profit. Usaha memaksimalkan profit ini akan diperoleh dari kombinasi pola pemotongan optimal dengan meminimalkan sisa bahan.

Dalam tugas akhir ini permasalahan knapsack dua dimensi diselesaikan dengan sebuah teknik heuristik. Dari analisa struktural knapsack dua dimensi diturunkan **algoritma aproksimasi**. Algoritma ini menggunakan permasalahan knapsack satu dimensi untuk menyelesaikan pemotongan bahan. Algoritma menggunakan pola-pola untuk menghasilkan sekumpulan strip-strip terbaik dan mengisikannya ke dalam lembaran bahan sehingga diperoleh kombinasi strip optimal. Algoritma ini dinamakan dengan algoritma **Best Strip Cutting (BSC)**. Dengan menggunakan metode program dinamis, jumlah knapsack yang terlibat dapat dikurangi. Karena profit dari pola adalah luasan dari pola itu sendiri, maka fungsi tujuan adalah meminimalkan sisa bahan yang tidak terpakai.

Algoritma Best Strip Cutting ini memberikan persentase solusi optimal 97%. Hasil uji coba terhadap data ukuran bahan yang besar menunjukkan efisiensi dari algoritma ini sehingga dapat digunakan untuk penyelesaian permasalahan di dunia nyata. Di bidang usaha industri yang berhubungan dengan permasalahan pemotongan bahan seperti kertas, kaca, plastik, plat logam dapat memanfaatkan penyelesaian ini dalam usaha mendapatkan profit maksimal dengan sisa bahan seminimal mungkin.

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Segala puji syukur penulis panjatkan ke hadirat Allah SWT yang Maha Pengasih lagi Maha Penyayang, karena hanya dengan rahmat dan taufiq-Nyalah pada akhirnya penulis dapat menyelesaikan tugas akhir dengan judul:

"PENYELESAIAN MASALAH KNAPSACK

DUA DIMENSI DENGAN

ALGORITMA APROKSIMASI"

Tugas akhir ini mempunyai bobot kredit 6 SKS, yang merupakan sebagian persyaratan untuk meraih gelar Sarjana Komputer di Jurusan Teknik Informatika - Fakultas Teknologi Industri - Institut Teknologi Sepuluh Nopember Surabaya.

Harapan penulis semoga Tugas Akhir ini dapat diterima dan bermanfaat bagi semua pihak yang membacanya.

Surabaya, 25 Januari 1999

Penulis

UCAPAN TERIMA KASIH

Dengan selesainya Tugas Akhir ini penulis ingin menyampaikan rasa terimakasih atas segala bantuan yang tak ternilai harganya kepada:

1. **Dr. Ir. Arif Djunaidy, Ph.D.**, selaku Ketua Jurusan Teknik Informatika ITS dan Dosen Pembimbing I.
2. **Ir. Esther Hanaya, M. Sc.**, selaku Sekretaris Jurusan Teknik Informatika ITS dan koordinator Tugas Akhir (*trimakasih yang dalam untuk nasehat-nasehatnya*)
3. **Dr. Ir. Drs.Ec. Ryanarto Sarno**, selaku Dosen Wali.
4. **Ir. Aries Tjahyanto, M. Kom.**, selaku Dosen Pembimbing II.
5. **Dosen-dosen Jurusan Teknik Informatika ITS**, yang telah menurunkan ilmunya dengan tulus ikhlas.
6. **Staf TU Jurusan Teknik Informatika ITS**, yang telah turut membantu memperlancar studi.
7. Rekan-rekan C06; **Retno, Nana, Nanik, Ninok, Syaiful, Andi, Ketut, Nico, Bahtiar, Pramu** dan khususnya **Mustakim** untuk segala support yang diberikan dengan tulus dan semoga Allah memberi balasan yang lebih baik.
8. Rekan-rekan C07-8; **Atiek, Entin, Umi dan Lailil**.
9. Teman-teman baikku; **Ijal, Wati, Lisa, M'Tantri** (*atas segala saran, nasehat dan bantuan yang takkan terlupakan, hanya Allah yang akan membalasnya dengan lebih baik*).
10. Rekan-rekan PMKTR; **Dedy, Iwan, Salman, Agus, Syahrul, B'Win dll**.
11. **Zulfa (Buby), D'Ama, D'Da, Qotie, KeRen, Nini, Icut, Andre** teman-temanku di KLANom serta **Mbo' Lin**.
12. **Untuk semua yang tak mungkin penulis sebutkan satu persatu.**

DAFTAR ISI

	halaman
ABSTRAK	iv
KATA PENGANTAR	v
UCAPAN TERIMA KASIH	vi
DAFTAR ISI	vii
DAFTAR GAMBAR	x
DAFTAR TABEL	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Permasalahan	2
1.3 Batasan Masalah	3
1.4 Tujuan dan Manfaat	3
1.5 Metodologi Penelitian	4
1.6 Sistematika Pembahasan	5
BAB II PERMASALAHAN KNAPSACK	7
2.1 Pengertian Masalah Knapsack	7
2.2 Macam-Macam Masalah Knapsack	9
2.3 Metode Penyelesaian Masalah Knapsack	12
2.3.1 Program Linear	12
2.3.2 Metoda Heuristik	13
2.4 Program Dinamis	15
2.5 Permasalahan Knapsack Constrained Multiple Choice	19
BAB III ALGORITMA APROKSIMASI	22
3.1 Penurunan Algoritma Aproksimasi	22
3.2 Algoritma Best Strips Cutting (BSC)	32
3.2.1 Identifikasi Masalah	32

3.2.2	Metode Pemotongan Bahan Dua Dimensi	34
3.3.4.1	Orthogonal dan Non-Orthogonal	34
3.2.4.2	Guillotine dan Non-Guillotine	34
3.2.3	Trimming	35
3.2.4	Efek Simetris	36
3.2.5	Efek Susunan Potongan	36
3.2.6	Normalisasi Pemotongan	37
3.2.7	Pembagian Plat Menjadi Sub Plat	38
3.2.8	Pembentukan Strip	40
3.2.8.1	Strip-Strip Horisontal	40
3.2.8.2	Strip-Strip Vertikal	41
3.2.9	Penempatan Strip-strip Pada Bahan Dua Dimensi	41
3.2.10	Fase 0-cut	46

BAB IV PERANCANGAN DAN PEMBUATAN

PERANGKAT LUNAK

48

4.1	Gambaran Umum Perangkat Lunak	48
4.2	Diagram Alir Data (DAD)	49
4.3	Perancangan Data	51
4.3.1	Data Masukan	51
4.3.2	Data Saat Pemrosesan	52
4.3.3	Data keluaran	54
4.4	Perancangan Proses	55
4.4.1	Pembentukan P_P dan P_L	55
4.4.2	Pembentukan Strip	57
4.4.3	Pengisian Plat	59
4.4.4	Pola Pemotongan	60
4.4.5	Hirarki Modul	61
4.5	Perancangan Menu	62
4.6	Pembuatan Perangkat Lunak	64
4.6.1	Struktur Data	64
4.6.2	Modul Pembentukan Strip	65
4.6.3	Modul Pengisian Plat	66
4.6.4	Modul Pembuatan Gambar	68



BAB V HASIL UJI COBA DAN PEMBAHASAN	69
5.1 Data Random	69
5.1.1 Hasil Komputasi	70
5.2 Data Terpilih	72
5.2.1 Penyelesaian Dengan BSC	74
5.2.2 Fase 0_Cut	77
5.2.3 Hasil Komputasi	78
 BAB VI PENUTUP	 80
6.1 Kesimpulan	80
6.2 Saran	81
 DAFTAR PUSTAKA	 82
 LAMPIRAN A: Panduan Pengoperasian Perangkat Lunak	 A-1
LAMPIRAN B: Data Masukan Random	B-1
LAMPIRAN C: Contoh Penyelesaian Secara Manual	C-1

DAFTAR GAMBAR

	halaman
Gambar 3.1 Pola Pemotongan Yang Diselesaikan Dengan Algoritma Aproksimasi	26
Gambar 3.2 Tiga Pola Pemotongan Optimal (pola w memiliki panjang terkecil masuk ke dalam strip horisontal tepat 3 kali)	31
Gambar 3.3a Pola Pemotongan Optimal Disusun Oleh 2 Sub Plat (3,11) Dan (6,11)	32
Gambar 3.3b Struktur Strip Ekuivalen	32
Gambar 3.4a Pola Pemotongan Yang Tidak Feasibel	35
Gambar 3.4b Pola Pemotongan Yang Feasibel Dengan Pemotongan Guillotine	35
Gambar 3.5a Pola Pemotongan Non-Normalisasi	38
Gambar 3.5b Pola Pemotongan Yang Di-Normalisasi	38
Gambar 4.1 DAD Level 0, Sistem Perangkat Lunak	49
Gambar 4.2 DAD Level 1, Proses Pencarian Pola Pemotongan	50
Gambar 4.3 Model Data Plat (P, L), Pola (p1, l1) dan (p2, l2)	51
Gambar 4.4 Diagram Alir Pembentukan P_p Dan P_L	56
Gambar 4.5 Diagram Alir Pembentukan Strip-Strip	58
Gambar 4.6 Diagram Alir Pengisian Plat	59
Gambar 4.7 Diagram Alir Pencarian Pola Pemotongan	60

Gambar 4.8	Hirarki Modul Sistem	61
Gambar 4.9	Menu Sistem Perangkat Lunak	62
Gambar 5.1	Solusi Feasibel 1 Untuk Plat (9,11)	74
Gambar 5.2	Solusi Feasibel 2 Untuk Plat (9,11)	75
Gambar 5.3	Solusi Feasibel 3 Untuk Plat (9,11)	75
Gambar 5.4	Solusi Feasibel 4 Untuk Plat (9,11)	76
Gambar 5.5	Solusi Feasibel 5 Untuk Plat (9,11)	77
Gambar 5.6	Solusi Feasibel Fase 0-Cut	77



MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER

DAFTAR TABEL

	halaman
Tabel 2.1 Pemuatan Kendaraan Yang Berkapasitas 10 Ton	15
Tabel 2.2 Nilai-Nilai Untuk $f_1()$ Dan $f_2()$	17
Tabel 2.3 Nilai-Nilai Untuk $f_1()$ Sampai $f_4()$	18
Tabel 2.4 Pengaruh Pemilihan Banyaknya Item Pada $f()$	19
Tabel 2.5 Data Input	20
Tabel 2.6 Output	21
Tabel 5.1 Hasil BSC Dan Fase 0-Cut (Ukuran Kecil)	70
Tabel 5.2 Hasil BSC Dan Fase 0-Cut (Ukuran Besar)	71
Tabel 5.3 Contoh 1	72
Tabel 5.4 Contoh 2	73
Tabel 5.5 Contoh 3	73
Tabel 5.6 Contoh 4	73
Tabel 5.7 Contoh 5	73
Tabel 5.8 Solusi Feasibel Pada Guillotine Vertikal	74
Tabel 5.9 Solusi Feasibel Pada Guillotine Horisontal	76
Tabel 5.10 Hasil BSC Dan Fase 0-Cut	78

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Dalam kehidupan sehari-hari sering dijumpai bahan/material dalam bentuk lembaran atau dua dimensi misalnya; tekstil, kaca, lembaran logam, kayu, kertas, plastik. Bahan-bahan dihasilkan oleh pabrik ini biasanya dalam ukuran besar, sehingga belum dapat dipergunakan/dimanfaatkan langsung. Supaya dapat memenuhi kebutuhan setiap konsumen yang berbeda-beda, bahan ini harus mengalami proses pemotongan terlebih dahulu. Pemotongan yang dilakukan harus dapat memenuhi kebutuhan dan memberi suatu keuntungan maksimal.

Permasalahan pemotongan dua dimensi yang tak terbatas adalah pemotongan selebar empat persegi panjang besar menjadi sejumlah empat persegi panjang kecil yang diberikan ukuran dan nilai sehingga dapat digunakan untuk memaksimalkan nilai bagian-bagian potongannya. Masalah ini kerap kali muncul sewaktu dilakukan pemotongan lembaran logam menjadi ukuran-ukuran yang diinginkan, dalam pemotongan karton untuk dibuat kotak dan dalam pemotongan kayu untuk membuat perabotan. Masalah pemotongan bahan ini adalah salah satu dari masalah knapsack. Sebuah knapsack akan diisi dengan sejumlah barang semaksimal mungkin sehingga sisa ruang seminimal mungkin.

Perkembangan teknologi menuntut pengembangan terus menerus dilakukan di bidang pemotongan bahan ini. Hal ini dipandang perlu dilakukan untuk

mengantisipasi semakin meningkatnya persaingan pasar di bidang industri. Penggunaan komputer dalam mengoptimalkan bahan semakin ditingkatkan sehingga pengaturan kombinasi pola pada suatu bahan dengan metode tertentu menjadi maksimal. Hasil yang dicapai akan dapat mempengaruhi biaya produksi akhir.

Dalam tugas akhir ini akan diajukan sebuah algoritma heuristik efisien untuk menyelesaikan permasalahan knapsack dua dimensi atau Two Dimensional Knapsack (TDK). Dari algoritma ini diturunkan *algoritma aproksimasi* yang menggunakan permasalahan knapsack satu dimensi atau One Dimensional Knapsack (ODK). Algoritma ini menggunakan serangkaian ODK untuk menghasilkan sekumpulan strip-strip optimal dan mengisikannya ke dalam lembaran bahan secara optimal menggunakan ODK yang lain. Untuk mengurangi jumlah ODK yang terlibat, metode program dinamis dapat dipakai. Jika profit dihubungkan dengan masing-masing pola dengan luas permukaannya, maka fungsi tujuan adalah meminimalkan sisa yang tidak terpakai.

1.2 PERMASALAHAN

Masalah knapsack yang akan diselesaikan dalam tugas akhir ini adalah masalah pemotongan benda dua dimensi tak terbatas. Bahan atau plat yang akan dipotong ini dianggap sebagai lembaran datar dengan ukuran besar berbentuk empat persegi panjang yang akan dipotong menjadi empat persegi panjang-empat persegi panjang berukuran kecil. Ukuran bahan A_0 dengan ukuran panjang dan lebarnya (P_0, L_0) akan dijadikan potongan empat persegi panjang kecil sebanyak m jenis pola dengan ukuran $\{(p_1, l_1), (p_2, l_2), \dots, (p_m, l_m)\}$.

Pengaturan letak pola-pola sebanyak m jenis harus dilakukan seproporsional mungkin untuk mendapatkan hasil yang optimal. Metode program dinamis digunakan untuk menyelesaikannya secara bertahap. Posisi pola-pola yang tepat akan dapat meminimalkan sisa bahan pula. Algoritma aproksimasi akan diterapkan untuk memberikan solusi optimal dari permasalahan pemotongan ini.

1.3 BATASAN MASALAH

Bahan yang akan dipotong dianggap umum (misal plastik, kaca, kertas, logam, atau tekstil) dengan jenis pemotongan guillotine. Memotong dengan guillotine berarti memotong mulai dari satu sisi sampai ke sisi satunya. Dengan pemotongan ini akan didapatkan hasil yang terpisah dari satu kali pemotongan. Pemotongan guillotine ini dilakukan dengan normalisasi pemotongan yaitu setiap pola yang dihasilkan letaknya bersisian dengan sisi pemotongan. Ukuran akibat pemotongan dianggap sangat tipis.

Banyaknya masing-masing pola persegi panjang yang akan dipotong adalah tak terbatas (*unconstrained*). Sehingga untuk satu macam pola dapat ditempatkan pada lembaran bahan lebih dari 1 kali. Nilai dari sebuah pola diasumsikan dengan luas dari pola itu sendiri ($p_i l_i$).

Arah pola tetap selama dalam pemotongan, sehingga (l, p) tidak sama dengan (p, l) .

1.4 TUJUAN DAN MANFAAT

Tujuan pembuatan tugas akhir ini adalah menyelesaikan masalah knapsack dua dimensi dengan algoritma aproksimasi. Algoritma ini mengoptimalkan

pemotongan benda dua dimensi tak terbatas (*unconstrained*) untuk memperbaiki hasil yang dicapai dengan metode sebelumnya (guillotine tanpa normalisasi). Solusi optimal yang diperoleh akan memaksimalkan nilai profit dengan meminimalkan sisa bahan yang tidak terpakai.

Penerapan algoritma aproksimasi untuk penyelesaian permasalahan knapsack dua dimensi ini dalam kehidupan nyata, dapat dimanfaatkan oleh siapa saja yang berhubungan dengan masalah pemotongan bahan dua dimensi untuk memaksimalkan profit dengan meminimalkan sisa bahan terbuang.

1.5 METODOLOGI PENELITIAN

Dalam tugas akhir ini telah dilakukan beberapa langkah yaitu:

- **Studi literatur** terhadap permasalahan knapsack khususnya pemotongan bahan baik lewat buku maupun jurnal yang memberikan suatu gambaran apa yang telah diperoleh dalam penyelesaian terhadap permasalahan selama ini. Karena tidak adanya metode khusus yang dapat digunakan untuk memberikan suatu hasil optimal maka metode heuristik tetap dipergunakan untuk menyelesaikan masalah ini. Diturunkannya algoritma aproksimasi dari analisa struktural pada metode heuristik yang digunakan menyelesaikan masalah knapsack dua dimensi ini, kemudian diterapkan dalam pemotongan bahan dua dimensi tak terbatas ini.
- **Perancangan sistem** dengan mempelajari, merancang, menguji serta menganalisa sistem

- **Pembuatan perangkat lunak** untuk mengimplementasikan algoritma-algoritma ke dalam program dan menggabungkannya menjadi sebuah sistem sesuai rancangan yang diinginkan.
- **Evaluasi** terhadap perangkat lunak yang telah diimplementasikan dengan menguji dan melakukan perbaikan dimana diperlukan yang pada akhirnya disusun menjadi laporan tugas akhir.

1.6 SISTEMATIKA PEMBAHASAN

Untuk memberikan gambaran yang jelas mengenai perancangan perangkat lunak yang dibuat maka buku tugas akhir ini disusun sebagai berikut:

Bab I **Pendahuluan**, dipaparkan tentang latar belakang masalah, permasalahan, batasan masalah tersebut serta tujuan dan manfaat dari tugas akhir ini.

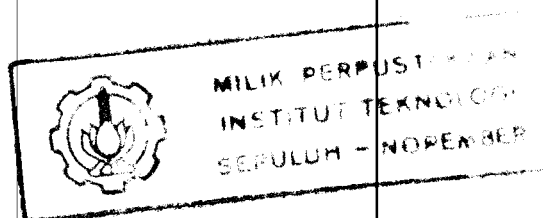
Bab II **Permasalahan Knapsack**, dijelaskan tentang apa itu masalah knapsack serta teori-teori yang mendasari dan menunjang untuk menyelesaikan suatu permasalahan knapsack.

Bab III **Algoritma Aproksimasi**, dijelaskan dari analisa struktural dari bahan dua dimensi berbentuk empat persegi panjang diturunkan algoritma aproksimasi yang diterapkan dalam pemotongan bahan ini meliputi identifikasi masalah penyusunan model, penyelesaian serta langkah-langkahnya dengan menggunakan algoritma tersebut.

Bab IV **Perancangan dan Pembuatan Perangkat Lunak**, dibahas tentang sistem yang akan dibuat, input dan tahap-tahap untuk mendapatkan output dari sistem.

Bab V **Hasil Uji Coba dan Pembahasan**, dilakukan uji coba dan evaluasi terhadap sistem yang telah diimplementasikan, kelebihan dan kekurangan dari sistem yang diimplementasikan.

Bab VI **Penutup**, disampaikan kesimpulan dari hasil uji coba yang telah dilakukan dan saran dan kemungkinan pengembangan berikutnya yang dapat dilakukan terhadap perangkat lunak.



BAB II

PERMASALAHAN KNAPSACK

Permasalahan knapsack terdiri dari beberapa macam. Jika ditinjau dari segi jumlah knapsack maka terdapat knapsack tunggal dan knapsack banyak. Jika ditinjau dari segi jumlah item maka terdapat permasalahan knapsack 0-1, permasalahan knapsack multiple choice. Dan jika ditinjau dari segi dimensinya terdiri dari permasalahan knapsack 1 dimensi, permasalahan knapsack 2 dimensi dan permasalahan knapsack 3 dimensi.

Pendekatan-pendekatan yang pernah dilakukan untuk penyelesaian terhadap permasalahan-permasalahan knapsack itu adalah dengan program linear dan heuristik. Untuk mengurangi jumlah knapsack yang dilibatkan dalam penyelesaian permasalahan knapsack ini digunakan program dinamis.

2.1 PENGERTIAN PERMASALAHAN KNAPSACK

Ketika melakukan suatu perjalanan liburan atau bisnis, maka akan ada batasan dalam jumlah barang yang akan dibawa. Demikian pula jika ingin pergi hiking, ransel yang digunakan hanya akan dapat memuat beberapa barang dengan volume yang terbatas. Jika mengadakan perjalanan dengan pesawat, berat (juga mungkin volume) barang yang dibawa dibatasi supaya tidak mendapat biaya tambahan. Apapun jenis kendaraan yang dipilih baik melalui jalan raya atau kereta api, akan selalu ada batasan jumlah barang yang dapat dibawa serta. Sebagai akibatnya seseorang harus memilih barang-barang mana saja yang akan

dibawa dan mana yang tidak. Idealnya, yang dibawa adalah barang-barang penting saja.

Dalam bidang usaha komersil, terdapat contoh-contoh permasalahan yang pada prinsipnya sama dengan seseorang yang akan mengadakan perjalanan, yaitu pembatasan jumlah pemuatan barang ke dalam ruang yang terbatas volumenya. Misalnya paket pengangkutan internasional yang mana membutuhkan pengepakan barang ke dalam kontainer. Jika paket ini pengirimannya menggunakan jasa pengangkutan udara, maka akan mendapat pembatasan yang ketat dalam berat total kontainer tersebut. Jasa pengangkutan laut juga membatasi berat total kargo dari barang-barang yang dipaket.

Masalah-masalah seperti di atas mulai diselidiki penyelesaiannya dengan program dinamis. Hal yang menjadi pertimbangan dengan keputusan ini adalah biaya atau profit. Disini titik perhatian pada masalah pengepakan dimana keputusan yang diambil melibatkan pemilihan sejumlah barang. Sekali lagi masing-masing keputusan yang dibuat akan membutuhkan biaya atau keuntungan yang berhubungan dengannya. Tapi keputusan yang dibuat akan mengurangi keterbatasan suplai dari beberapa resource seperti berat, biaya dan volume. Keputusan mungkin sederhana apakah akan mengambil beberapa item atau tidak; alternatifnya, akan memperhatikan jumlah resource yang akan diambil. Masalah inilah yang telah diuraikan diatas ini disebut dengan *permasalahan knapsack*.

Jadi permasalahan knapsack adalah permasalahan ketika seseorang yang akan mengadakan perjalanan dengan membawa sebuah kantong. Kantong ini dapat diisi dengan bermacam-macam barang yang memiliki nilai masing-masing. Masalah muncul ketika harus memilih barang mana yang harus dibawa sehingga sesuai dengan kapasitas kantong tadi.

Dari gambaran diatas dapat dirumuskan menjadi suatu bentuk matematis. Jika n adalah jumlah item barang, p_j adalah nilai barang ke j , x_j adalah jumlah barang ke j yang dipilih, w_j volume item ke j , dan c adalah kapasitas volume kantong.

Perumusan matematika dari persoalan di atas:

$$\text{Fungsi tujuan: } \sum p_j x_j \quad (2.1)$$

$$\text{Syarat: } \sum w_j x_j \leq c, x_j \geq 0, j = 1, \dots, n$$

2.2 JENIS PERMASALAHAN KNAPSACK¹

Permasalahan knapsack ada bermacam-macam tergantung dari mana peninjauan yang dilakukan terhadap masalah tersebut.

Jika dilihat berdasarkan jumlah pemakaian knapsack, maka permasalahan knapsack dibedakan menjadi dua macam sebagai berikut:

- Permasalahan knapsack tunggal, yaitu permasalahan knapsack yang memiliki hanya satu buah knapsack yang akan diisi. Persamaan matematikanya menjadi:

$$\text{Maksimum: } \sum p_j x_j \quad (2.2)$$

$$\text{Syarat: } \sum w_j x_j \leq c, 0 \leq x_j \leq b_j,$$

$$x_j \text{ adalah integer, } j = 0, \dots, n$$

- Permasalahan knapsack banyak, yaitu permasalahan knapsack yang memiliki lebih dari satu buah knapsack yang akan diisi. Persamaan matematikanya menjadi:

¹ Mathello S., Toth P., "Knapsack Problems; Algorithms and Computer Implementations", John Wiley & Sons, 1990



$$\text{Maksimum: } \sum p_j x_j \quad (2.3)$$

$$\text{Syarat: } \sum w_j x_j \leq c, x_j \geq 0,$$

x_j adalah integer, $j = 0, \dots, n$

Jika dilihat berdasarkan jumlah item yang tersedia, maka dapat dibedakan sebagai berikut:

- Permasalahan knapsack 0-1, yaitu jumlah item dari masing-masing barang yang dapat masuk ke dalam knapsack hanya satu buah atau tidak diambil. Persamaan matematikanya menjadi:

$$\text{Maksimum: } \sum p_j x_j \quad (2.4)$$

$$\text{Syarat: } \sum w_j x_j \leq c, x_j = 0 \text{ atau } 1$$

x_j adalah integer, $j = 0, \dots, n$

- Permasalahan knapsack Multiple-Choice, yaitu jumlah item dari masing-masing barang dapat lebih dari satu buah atau tidak samasekali.

Persamaan matematikanya menjadi:

$$\text{Maksimum: } \sum p_j x_j \quad (2.5)$$

$$\text{Syarat: } \sum w_j x_j \leq c, 0 \leq x_j \leq b_j,$$

x_j adalah integer, $j = 0, \dots, n$

Kemudian jika permasalahan knapsack ditinjau dari jumlah item yang dioptimalkan, dibedakan menjadi dua macam terdiri dari:

- Permasalahan Knapsack Bounded, yaitu jumlah setiap item yang dipilih untuk masuk ke dalam knapsack dibatasi sesuai order/ permintaan.
- Permasalahan Knapsack Unbounded, yaitu jumlah setiap item yang dipilih tidak terbatas jumlahnya.

Jika permasalahan knapsack ditinjau dari dimensi benda maka dapat dibedakan menjadi tiga macam sebagai berikut:

- Permasalahan Knapsack 1 Dimensi, yaitu peninjauan hanya pada dimensi panjang saja. Misalnya kabel, log kayu dan batang besi.
- Permasalahan Knapsack 2 Dimensi, yaitu peninjauan hanya pada dimensi panjang dan lebar. Misalnya kertas, tekstil, kaca dan plate besi.
- Permasalahan Knapsack 3 Dimensi, yaitu peninjauan hanya pada dimensi panjang, lebar dan tinggi. Misalnya kayu dan pengisian mobil boks.

Namun secara umum pada dasarnya permasalahan knapsack ini ada 2 macam, yaitu;

- Mengisi ruang, contohnya pengisian kontainer dengan boks dengan berbagai ukuran sehingga dapat memuatnya semaksimal mungkin.
- Membagi ruang, contoh bahan persegi yang berukuran besar dibagi-bagi (dipotong) sehingga menjadi bagian-bagian kecil untuk memberi nilai profit yang besar.

2.3 METODE PENYELESAIAN PERMASALAHAN KNAPSACK

Permasalahan knapsack yang ada telah diselesaikan dengan pendekatan metode program linier dan heuristik.

2.3.1 Program Linear

Permasalahan knapsack dapat dirumuskan sebagai sebuah program linear² dengan tujuan untuk meminimumkan sisa. Rumus matematika untuk linier programming terdiri dari n kolom yang berkorespondensi dengan sebuah cutting pattern. Pada masing-masing iterasi dari linear program, metode simpleks memilih sebuah kolom dan sebuah variabel untuk masuk sebagai basis. Rumus matematikanya didefinisikan sebagai berikut:

$$\text{Min:} \quad x_0 = c_1x_1 + c_2x_2 + \dots + c_jx_j + c_nx_n \quad (2.6)$$

$$\text{Syarat:} \quad a_{i1}x_1 + a_{i2}x_2 + \dots + a_{ij}x_j + a_{in}x_n \geq N_i$$

$$\text{dimana:} \quad i = 1, 2, \dots, m \text{ dan } x_j \geq 0$$

Dalam notasi matriks dinyatakan dengan :

$$Ax = b, \text{ dimana:}$$

$$A = \begin{bmatrix} 1 & -c_1 & -c_2 & \dots & -c_n \\ 0 & a_{11} & a_{12} & \dots & a_{1n} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}, x = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_m \end{bmatrix}, b = \begin{bmatrix} 0 \\ N_1 \\ \vdots \\ N_m \end{bmatrix}$$

Gilmore dan Gomory telah melakukan riset tahun 1961, memikirkan permasalahan knapsack ini sebagai masalah dengan 2 tahap penyelesaiannya yaitu

² R. G. Dyson, A. S. Gregory, 'The Cutting Stock Problem in the Flat Glass Industry', *Operational Research Quarterly* Vol. 25 No.1, hal 50-51

knapsack problem yang diselesaikan sebagai knapsack problem satu dimensi dan tahap kedua sebagai masalah linear programming.

2.3.2 Metode Heuristik

Ada beberapa definisi Heuristik yang dipergunakan dalam riset operasional:

- Metode heuristik dipergunakan saat tidak ditemukannya algoritma yang dengan cara iterasi dapat menghasilkan suatu solusi optimal atau cukup baik.
- Metode berdasarkan pengalaman telah memperlihatkan hasil solusi yang baik tapi tidak menjamin solusi tersebut maksimum.
- Cara memecahkan masalah dengan pendekatan intuisi.
- Simulasi dari bagaimana manusia dapat mencari solusi yang baik dari setiap masalah

Metode heuristik diambil jika untuk suatu permasalahan tidak ada suatu algoritma eksak yang dapat menyelesaikannya. Pendekatan heuristik akan dilakukan jika:

1. Heuristik dapat menghasilkan suatu solusi yang mendekati optimal.
2. Heuristik cukup fleksibel dengan menghasilkan beberapa solusi sub-optimal.
3. Cepat waktu komputasinya, karena membatasi pencarian dengan mengurangi jumlah alternatif yang ada.
4. Mudah diterapkan, mudah dimengerti oleh pemakai
5. Kebutuhan penyimpanan cukup kecil

Dalam penyelesaian permasalahan knapsack digunakan metode heuristik³ untuk mencari solusi optimumnya dengan memilih alternatif-alternatif solusi yang paling optimal. Dalam permasalahan knapsack tidak ada suatu algoritma yang eksak yang dapat menghasilkan solusi maksimal atau terbaik. Metode heuristik memecahkan permasalahan knapsack dengan hasil yang optimal. Metode heuristik dinyatakan dalam bentuk matematis yang disesuaikan dengan masalah untuk menuju ke solusi optimum yang memuaskan bagi masalah tersebut.

Strategi yang dipertimbangkan jika tidak ada metode efisien untuk menemukan solusi optimal sebagai berikut:

1. Mencari algoritma yang efisien.
2. Hanya kasus khusus dari masalah yang diperhatikan.
3. Melonggarkan beberapa pembatas dan mengembangkan algoritma untuk masalah itu.
4. Rancang algoritma yang bekerja cepat pada banyak masalah yang dihadapi.
5. Menetapkan metode yang bekerja cepat dan menghasilkan sesuatu yang dapat dipergunakan yang hasilnya mendekati optimal.

Program dinamis sering digunakan dalam penyelesaian permasalahan knapsack untuk mengurangi jumlah knapsack yang terlibat. Di sini akan dijelaskan lebih rinci tentang program ini karena dalam tugas akhir ini juga menggunakannya.

³ J.E. Beasley, "Algorithms for Unconstrained Two-Dimensional Guillotine Cutting", J. Opl. Res. Soc. Vol 36., No. 4, hal 303-304, 1985

2.4 PROGRAM DINAMIS

Keputusan yang diambil oleh program dinamis adalah hasil dari keputusan-keputusan sebelumnya. Dapat dikatakan bahwa program dinamis membagi masalah menjadi sub-sub bagian yang lebih kecil dengan keputusan akhir yang dibuat merupakan hubungan keputusan-keputusan dari sub-sub bagian tadi. Program dinamis dipergunakan dalam menyelesaikan permasalahan knapsack dalam usaha mencapai hasil optimal dari penyelesaian masalah.

Rumus matematika untuk program dinamis harus dikembangkan agar memenuhi kondisi dari permasalahan knapsack yang akan diselesaikan. Hal ini dikarenakan program dinamis tidak mempunyai bentuk standar seperti linear programming. Berikut contoh permasalahan knapsack dengan menggunakan program dinamis untuk penyelesaiannya.

1. Permasalahan Knapsack Sederhana (dimensi 1)

Sebuah kendaraan dengan kapasitas 10 ton dan tersedia 4 item barang yang akan dimuat ke kargo; berat dan nilai (banyaknya) ditunjukkan pada tabel 2.1. Bagaimana seharusnya kapal dimuati supaya jumlah barang maksimal dalam kargo ?

Tabel 2.1 Pemuatan kendaraan yang berkapasitas 10 ton

Nomor barang (i)	1	2	3	4
Berat ton (W_i)	2	3	4	6
Nilai (V_i)	8	12	15	19

Masalah ini dapat ditulis sebagai fungsi dengan tujuan akan dmaksimalkan:

Maks: $f(x) = 8x + 12x + 15x + 19x$

dimana syarat : $2x_1 + 3x_2 + 4x_3 + 6x_4 \leq 10$

$x_1, x_2, x_3, x_4 = 0$ atau 1

Masalah ini dapat diselesaikan dengan cara program dinamis. Pertama, masalah dipecahkan menjadi beberapa sub masalah yang sederhana. Pada masing-masing tahap harus diputuskan apakah produk akan diikuti atau tidak. Untuk program dinamis perlu didefinisikan kondisi dan tahap-tahap; dimana tahap-tahap dari pemuatan barang berhubungan dengan jumlah keputusan yang masih harus dibuat. Pertama ada empat keputusan, lalu tiga sampai seluruh keputusan dilakukan. Informasi yang mencukupi untuk menerangkan kondisi saat mengambil keputusan-keputusan ini akan menjadi jumlah ruang tersisa di kendaraan.

Ada 10 ton yang tersedia pada mulanya, 10 ton ini akan berkurang setelah keputusan ke-1 diambil; dan akan berkurang sampai keputusan terakhir.

Asumsi ada n tahap dan kapasitas T ton dalam kendaraan. Jika didefinisikan $f(T)$ sebagai nilai muatan optimal masing-masing kondisi dan tahap, maka:

$$\begin{cases} f_n(T) = \max\{f_{n-1}(T), V_n + f_{n-1}(T - W_n)\} & \text{untuk } (T \geq W_n) \\ f_n(T) = f_{n-1}(T) & \text{untuk } (T < W_n) \\ f_0(T) = 0 & \text{untuk } \forall T \end{cases} \quad (2.7)$$

Maksudnya, yang paling baik dilakukan pada tahap n dan dengan state T adalah tidak melakukan pemuatan dari produk n ($f(T)$) dan loading produk n dan ke keputusan berikut dengan state berkurang beratnya ($f_{n-1}(T - W_n)$) dimana T cukup besar untuk dapat ini terjadi.

Jelasnya, $f(T)$ hanya akan memiliki 2 kemungkinan nilai: 0 atau 19; akan menjadi 0 jika T kurang dari 6 dan 19 jika T adalah 6 atau lebih. Dengan kata



MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER

lain saat tercapai pada tahap itu dibuat keputusan tentang produk nomor 4, yang akan dimuat jika tersedia cukup ruang untuk itu.

$$F_2(10) = \max \{f_1(10), 15 + f_1(6)\}$$

$$= \max \{19, 15 + 19\}$$

$$= 34$$

$$F_2(10) = \max \{f_1(9), 15 + f_1(5)\}$$

$$= \max \{19, 15 + 0\}$$

$$= 19$$

$$F_2(1) = f_1(1)$$

$$= 0$$

$$F_2(0) = 0$$

Nilai-nilai ini untuk $f_1()$ dan $f_2()$ dalam tabel 2.2.

Tabel 2.2 Nilai-nilai ini untuk $f_1()$ dan $f_2()$

T	1	2	3	4	5	6	7	8	9	10
$f_1(T)$	0	0	0	0	0	19	19	19	19	19
$f_2(T)$	0	0	0	0	15	15	19	19	19	34

Proses berpindah dari arah belakang (backward) ke proses keputusan, berikutnya dihitung $f_3()$ menggunakan $f_2()$ (tapi bukan $f_1()$). Logikanya sama dengan yang sebelumnya:

$$F_3(10) = \max \{f_2(10), 12 + f_2(7)\}$$

$$= \max \{34, 12 + 19\}$$

$$= 34$$

$$F_3(9) = \max \{f_2(9), 12 + f_2(6)\}$$

$$= \max \{19, 12 + 19\}$$

$$= 31$$

$$F_3(1) = f_2(1)$$

$$= 0$$

$$F_3(0) = 0$$

Dengan meneruskan cara ini akan didapat nilai $f_4()$. Seperti ditunjukkan pada tabel 2.3. T mengambil nilai dari 0 sampai 11 (tidak sesuai dengan kapasitas kapal 10). Penambahan ini hanya untuk mengecek sensitivitas jawaban sesuai dengan kapasitas kapal.

Diberikan tambahan satu unit kapasitas, berapa keuntungan yang akan diberikan untuk pemuatan yang optimal. Jawabannya adalah akan meningkatkan nilai unit 4 dari 35 menjadi 39.

Tabel 2.3 Nilai-nilai ini untuk $f_1()$ dan $f_2()$

T	0	1	2	3	4	5	6	7	8	9	10	11
$f_1(T)$	0	0	0	0	0	0	19	19	19	19	19	19
$f_2(T)$	0	0	0	0	15	15	15	19	19	19	34	34
$f_3(T)$	0	0	0	12	15	15	19	27	27	31	34	34
$f_4(T)$	0	0	8	12	15	20	23	27	27	35	35	39

Nilai optimal akan menjadi 35 unit jika kapal dipak dengan cara optimal.

2. Pengembangan Rumus

Ada suatu keterbatasan dengan menggunakan satu item dari masing-masing barang dalam kapal. Gambaran berikut ini sama dengan yang pertama, tapi keputusan akan lebih kompleks: disamping menggunakan biner (0 atau 1), nomor barang item hanya dibatasi oleh ruang dalam kapal, seperti yang diukur oleh dari sisa kapasitas (T ton) dibagi berat (W_n ton) dari nya. Persamaan yang diperoleh:

$$f_n(T) = \max_{0 \leq i_n \leq \lfloor \frac{T}{W_n} \rfloor} \{ f_{n-1}(T), V_n + f_{n-1}(T - i_n W_n) \} \quad (2.8)$$

Kemungkinan baru akan membuat sedikit perbedaan dalam perhitungan keputusan optimal; mulai dengan satu jenis barang, dan berjalan terus dengan dua jenis barang dan seterusnya. Hasilnya berbeda secara umum, karena ada beberapa state dan tahap dari mana keputusan optimal akan dipak lebih dari satu unit barang dan keputusan optimal akan berbeda dari yang lain seperti yang telah ditemukan pada batasan satu unit. Hasilnya pada tabel 2.4.

Tabel 2.4 Pengaruh pemilihan banyaknya item pada $f()$

T	0	1	2	3	4	5	6	7	8	9	10	11
$f_1(T)$	0	0	0	0	0	0	19	19	19	19	19	19
$f_2(T)$	0	0	0	0	15	15	19	19	19	19	34	34
$f_3(T)$	0	0	0	12	15	15	24	27	27	30	36	40
$f_4(T)$	0	0	8	12	15	20	24	28	32	36	40	44

Nilai-nilai yang diperoleh dalam tabel 2.4 tidak lebih kecil dari tabel 2.3 sebelumnya. Nilai dari keputusan optimal telah meningkat dari 35 ke 40 dan yang lainnya mengikuti dengan sendirinya. Perubahan diharapkan antara solusi masalah dengan keputusan biner dan dengan keputusan diskret menjadi positif, karena kemudian memiliki lebih sedikit batasan. Keputusan optimum $f_4(10)$ menghasilkan dengan 5 unit item 1, atau 2 unit item 2 dengan 2 unit item 1.

2.5 PERMASALAHAN KNAPSACK BOUNDED MULTIPLE CHOICE⁴

Permasalahan knapsack bounded (terbatas) ini diterapkan pada bahan kaca. Jumlah knapsack yang digunakan adalah tidak terbatas sehingga permasalahan knapsack ini merupakan permasalahan knapsack multiple-choice.

⁴ Pratono A., "Perancangan dan Pembuatan Perangkat Lunak Optimasi Pemotongan Benda Dua Dimensi", TG ITS, 1995

Permasalahan knapsack dua dimensi ini dipecahkan menjadi menjadi dua masalah berdimensi satu. Penyelesaian dilakukan mula-mula terhadap masalah satu dimensi pertama baru kemudian terhadap yang lain.

Aturan yang diterapkan pada pemotongan kaca disini adalah guillotine, karena kaca yang harus dipotong secara dipatahkan mengingat sifat kekerasan kaca tadi. Pemotongan bahan kaca ini menggunakan bantuan metode program dinamis yang digunakan untuk pemilihan cutting pattern dan alokasi cutting pattern.

Proses knapsack menghasilkan kombinasi penempatan benda ke dalam stok, yang mana stok berukuran (P, L) . Proses pertama menghasilkan penempatan ke salah satu sisi (satu dimensi pertama), dimana data stok disimpan. Hasil dari proses ini dipakai untuk proses terhadap knapsack kedua. Proses ini menjalani tiga tahap untuk mencapai nilai optimum yaitu: konversi data panjang dan lebar ke data ukuran (penyusunan ulang), mencari solusi feasibel dan mencari penempatan optimum.

Arah proses dipilih oleh si pemakai dengan memilih ke arah horisontal, vertikal atau kombinasi keduanya. Akhirnya dipilih yang paling optimum diantara hasil feasibel tersebut.

Tabel 2.5 Data input

No.	$p \times l$	b_i
0	2 x 3	20
1	3 x 4	22
2	3 x 5	27
3	2 x 7	15

Kapasitas stok = 20 x 14 dengan jumlah tidak terbatas.

Hasil akhir output yang dianalisa menunjukkan perbandingan dengan penyelesaian secara manual sebagai berikut:

Tabel 2.6 Output

No.	Uraian	Otomatis	Manual
1	Jumlah stok yang terpakai	5	5
2	Sisa yang ditinggalkan sampai stok yang ke n-1	0 satuan	307 satuan

BAB III

ALGORITMA APROKSIMASI

Dalam bab ini dijabarkan algoritma aproksimasi serta identifikasi permasalahan knapsack dalam pemotongan bahan, asumsi-asumsi yang digunakan sampai kepada langkah penyelesaian dengan algoritma tersebut. Langkah-langkah utama ini diterapkan dalam implementasi penyelesaian permasalahan knapsack dua dimensi .

3.1 PENURUNAN ALGORITMA APROKSIMASI

Algoritma aproksimasi ini diturunkan secara alami dari analisa struktural pada fase 0-cut (tidak mengalami pemotongan guillotine pada X_j). ODK yang ada diselesaikan secara terpisah dengan algoritma aproksimasi polinomial. Algoritma ini menyelesaikan r buah knapsack (r jumlah lebar yang berbeda) yang lebih kecil daripada m (jumlah macam pola yang akan dipotong) dan ukuran kedua adalah panjang plat P . Knapsack yang lain yang dibutuhkan adalah r dan ukuran kedua adalah lebar plat L .

Kemudian algoritma membutuhkan v buah knapsack (v jumlah panjang yang berbeda) dengan m ukuran maksimumnya dan ukuran kedua L . Knapsack satunya dengan ukuran v dan yang kedua adalah P . Sehingga knapsack yang dibutuhkan untuk diselesaikan adalah $(r + v + 2)$ buah knapsack.

Batas-batas dari algoritma aproksimasi ini akan dijelaskan lebih lanjut agar dapat meminimalkan sisa bahan yang tidak terpakai. Dalam hal ini batas-batas ini adalah upaya memaksimalkan nilai profit. Jadi dapat dikatakan usaha meminimalkan sisa bahan sama dengan memaksimalkan profit.

Lemma 1.

Sisa pada masing-masing strip yang dihasilkan oleh prosedur yang menghasilkan strip adalah $< (1/(k+1)) S_i$,

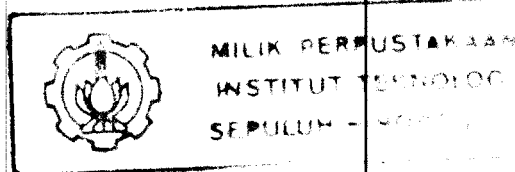
dimana:

S_i = luas strip,

$i = 1, 2, \dots, r$,

$k = \lfloor P/p \rfloor$, p adalah panjang pola paling lebar yang dapat masuk dalam strip-strip horisontal atau

$k = \lfloor L/l \rfloor$, l adalah lebar pola paling panjang yang dapat masuk ke dalam strip-strip vertikal.



Bukti:

$$P/(k+1) < p \lfloor P/k \rfloor \text{ sehingga, } k(P/(k+1)) < kp \leq k \lfloor P/k \rfloor \leq P.$$

Strip dengan lebar l_i memiliki nilai V_i yang memenuhi $V_i \geq kpl_i$ karena ada konstruksi yang memasukkannya paling sedikit k pola-pola dengan panjang p .

$$\text{Jadi, } V_i > k (P/(k+1)) l_i$$

$$= (k/(k+1)) Pl_i$$

$$= (k/(k+1)) S_i$$

Hal yang sama berlaku juga dalam pembuktian untuk strip-strip vertikal.

Corollary 1:

Rasio aproksimasi dari algoritma adalah mendekati $1/2$

Bukti:

Dari lemma 1 diperoleh untuk masing-masing strip $V_i > 1/2 S_i$.

Jadi nilai $V = \sum_{i \in E} 1/2 V_i$, dimana E adalah himpunan indeks strip-strip terpilih dari knapsack terakhir. Oleh karena itu;

$$\begin{aligned} V &= \sum_{i \in E} 1/2 V_i \\ \Leftrightarrow V &> \sum_{i \in E} 1/2 S_i \\ &= \sum_{i \in E} 1/2 P l_i \end{aligned}$$

Karena $L - \sum_{i \in E} 1/2 l_i < l^*$, dimana l^* adalah lebar paling kecil (minimum) dari pola-pola tersebut, sehingga didapatkan;

$$\Leftrightarrow V > \sum_{i \in E} 1/2 P(L - l^*)$$

Nilai profit $V^* \leq PL$ sehingga menjadi

$$\Leftrightarrow V/V^* > \frac{1}{2} (1 - l^*/L)$$

Pada saat panjang kecil ($l^* \ll$), rasio aproksimasi ini cenderung menjadi $1/2$.
(Terbukti).

Lemma 2.

- i.) Jika $p_i > 1/2 P$ untuk semua $i = 1, \dots, m$, maka solusi optimal ekuivalen dengan solusi yang disusun oleh sebuah strip vertikal.
- ii.) Jika $l_i > 1/2 L$ untuk semua $i = 1, \dots, m$, maka solusi optimal ekuivalen dengan solusi yang disusun oleh sebuah strip horisontal.
- iii.) Jika $k = \max\{P/p_i, L/l_i, i = 1, \dots, m\} = 1$ atau 2, maka solusi optimal disusun oleh strip-strip vertikal atau horisontal.
- iv.) Solusi yang diperoleh algoritma aproksimasi adalah $1 - \varepsilon$ untuk semua $\varepsilon > 0$.

Bukti:

- i.) Tidak ada satu pola pun dapat masuk $>$ satu kali ke dalam plat secara horisontal. Gambar 3.1a.
- ii.) Sama dengan atas, tidak ada satu pola pun dapat masuk $>$ satu kali ke dalam plat secara vertikal. Gambar 3.1b.
- iii.) $k = 1$ adalah trivial,

Untuk $k = 2$, terdapat pola dengan indeks w , sehingga;

$$1/3P < p_w \leq 1/2P, \quad 1/3P < p_i \leq P \text{ dimana } \forall i \neq w, \text{ dan}$$

$1/3L < l_i \leq L = 1, \dots, m$. Jadi paling sedikit satu pola masuk tepat 2 kali pada sisi panjang bahan atau,

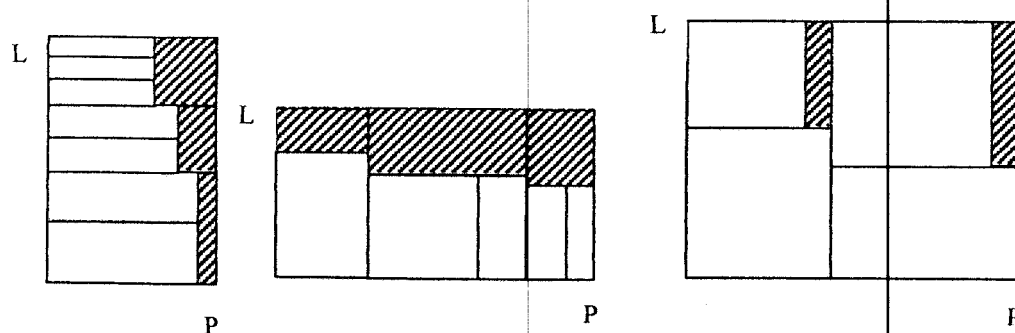
$$1/3L < l_w \leq 1/2L, \quad 1/3L < l_i \leq L \text{ dimana } \forall i \neq w, \text{ dan}$$

$1/3 P < p_i \leq P \quad \forall i = 1, \dots, m$. Jadi paling sedikit satu pola masuk tepat 2 kali pada sisi lebar bahan.

Asumsi $k = 2$ dihasilkan dari pola p_w . Pola pemotongan optimal akan dihasilkan pada pemotongan guillotine (vertikal) dimana pada masing-masing panjang sub plat, pola ini hanya masuk 1 kali saja. Jadi solusi optimal dihasilkan oleh 2 strip vertikal. Gambar 3.1c.

Jika pemotongan guillotine secara horisontal maka solusi optimal dihasilkan dari susunan 1 strip horisontal atau 2 strip horisontal.

- iv.) Masalah ODK yang diselesaikan adalah yang dapat menciptakan strip-strip horisontal dari panjang P dan strip-strip vertikal dari lebar L dengan memakai algoritma polinomial dalam $1-1/2\varepsilon$ dan dua ODK terakhir yang memilih strip terbaik yang dapat masuk ke dalam plat juga dalam range $1-1/2\varepsilon$.



(a). Struktur strip vertikal, (b). Struktur strip horisontal, (c). Struktur strip-strip vertikal.

Gambar 3.1. Pola pemotongan yang diselesaikan dengan algoritma aproksimasi dalam $1-1/2\varepsilon$.

Teori 1:

Jika V menyatakan nilai solusi yang dihasilkan algoritma aproksimasi dan V^* adalah nilai solusi optimal maka:

$$V > \frac{1}{2} (k / (k+1)) V^*, \text{ dimana } k = \max \left\{ \left\lfloor \frac{P}{L} \right\rfloor, \left\lfloor \frac{L}{P} \right\rfloor, i = 1, \dots, m \right\}$$

Bukti:

Misal w adalah pola dengan $k = \max \{ \lfloor \frac{p}{l_i} \rfloor \lfloor \frac{l}{l_i} \rfloor, i = 1, \dots, m \}$ dan asumsi k ini berkoresponden dengan panjang p_w . Strip horisontal memiliki lebar l_w .

Dengan lemma 1, nilai strip yang dihasilkan:

$$V_w > (k/(k+1)) PL_w.$$

Akibatnya algoritma menghasilkan:

$$V \geq \lfloor \frac{l}{l_w} \rfloor V_w > \frac{k}{k+1} P l_w \lfloor \frac{l}{l_w} \rfloor$$

dan karena $V^* \leq PL$, maka:

$$\frac{V}{V^*} > \frac{k}{k+1} \frac{l_w}{L} \lfloor \frac{L}{l_w} \rfloor \geq \frac{k}{k+1} \frac{\lfloor \frac{L}{l_w} \rfloor}{\lfloor \frac{L}{l_w} \rfloor + 1} \quad (3.1)$$

dan,

$$V > (\frac{1}{2} \frac{k}{k+1}) V^*$$

Teori 2:

Rasio aproksimasi algoritma adalah 4/9.

Bukti:

Dengan memperhatikan pola pemotongan optimal yang dinormalisasi dibuktikan bahwa, untuk $k = 3$ maka:

1. Ada pola w yang masuk 2 kali pada panjang plat dan pada lebarnya, atau
2. Solusi optimal disusun hanya oleh strip-strip vertikal atau horisontal.

Untuk $k = 3$ dan diasumsikan nilai ini didapat dari panjang p_w , didapatkan bahwa:

$$1/4 P < p_w \leq 1/3 P \quad (3.2)$$

yang mana,

$$1/4 P < p_i \leq P, \forall i \neq w \text{ dan } 1/4 L < l_i \leq L, \forall i = 1, 2, \dots, m$$

Jika $x_j \leq 1/2 P$, $x_j \in P_p$ adalah pemotongan guillotine yang pertama (arah vertikal) untuk pola pemotongan yang optimal. Pemotongan ini membagi plat awal menjadi 2 buah sub plat. Dua sub plat ini mengisi sebuah pola pada panjangnya atau satu pola pada sub plat pertama dan 2 pola pada yang kedua (atau sebaliknya).

Kasus pertama, solusi optimal disusun oleh dua strip vertikal. Dengan menerapkan lemma 2 algoritma memberikan solusi ini dalam $1 - \varepsilon$ untuk setiap $\varepsilon > 0$.

Kasus kedua, dianggap sebagai 2 sub kasus.

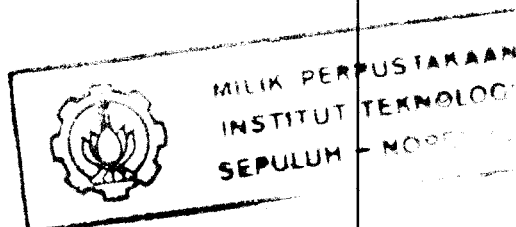
Pertama, dimisalkan:

$$1/3 L < l_i \leq L, \forall i = 1, 2, \dots, m \quad (3.3)$$

Di sini tidak ada pola yang dapat masuk secara vertikal lebih dari 2 kali. Dengan lemma 2 jelas bahwa solusi optimal dari sub plat kedua disusun oleh 2 strip vertikal atau 2 strip horisontal. Dengan demikian solusi optimal dari plat awal memasukkan:

- i.) Satu strip vertikal (untuk sub plat ke-1) dan 2 strip horisontal (sub plat ke-2), atau
- ii.) Tiga strip vertikal

Solusi terakhir disusun oleh tiga strip vertikal yang dihasilkan oleh algoritma (dalam $1 - \varepsilon$).



Untuk solusi yang lainnya algoritma memberikan worst case ratio aproksimasi adalah $4/9$. Pada kenyataanya seperti contoh solusi yang memiliki struktur gambar 3.2a. Pola pemotongan optimal memasukkan 3 kali pemotongan guillotine berturut-turut yang dibuat di titik X_j secara vertikal, di titik Y_j secara horisontal, dan di titik X_p secara vertikal. Dengan menganggap bahwa pola pemotongan guillotine normal memperhatikan efek-efek simetri, maka didapat:

$$x_j \leq \lfloor \frac{1}{2}P \rfloor, \quad y_j \leq \lfloor \frac{1}{2}L \rfloor, \quad x_p \leq \lfloor \frac{1}{2}(P - x_j) \rfloor$$

Strip-strip yang memasukkan pola w dan q dapat menjadi strip-strip trivial. Jika $l_p > 1/2 L$ diperoleh dari Lemma 1 dan memperhatikan 2 strip horisontal yang dihasilkan oleh pola p dan q sehingga,

$$V_p > 2/3 P l_p > 1/3 P L.$$

Dengan menggunakan kondisi 3.3, didapat,

$$V_q > 1/2 P l_q > 1/6 P L.$$

Akibatnya, solusi yang diperoleh:

$$V \geq V_p + V_q > 1/2 P L.$$

sehingga,

$$V > 1/2 V^*$$

Jika $l_p \leq 1/2 L$, maka pola p masuk 2 kali pada panjang dan 2 kali pada arah lebar. Dengan Lemma 1 dan kondisi 3.1, diperoleh:

$$V \geq V_p > 4/9 P L.$$

sehingga,

$$V > 4/9 V^*$$

Kedua, dianggap hanya ada sebuah pola p sehingga,

$$1/4 L < l_p \leq 1/3 L$$

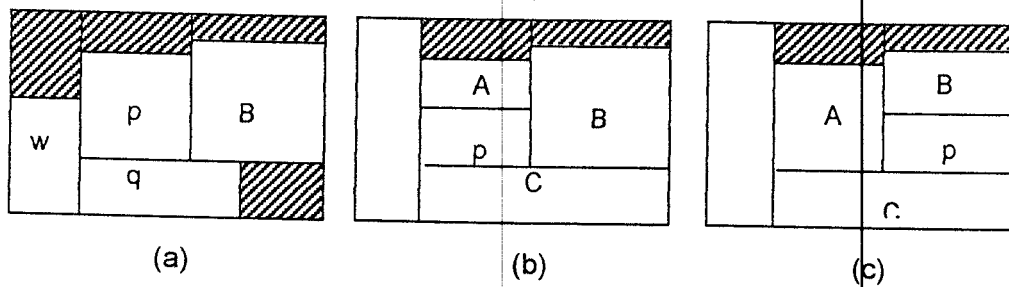
Ada sebuah pola dengan lebar l_p yang dapat masuk secara vertikal 3 kali ke dalam plat. Jika $p = w$ maka algoritma memberikan solusi $> 9/16$ solusi optimal (kondisi 3.1).

Sebaliknya situasi pada gambar 3.2b dan 3.2c akan muncul, jika salah satu dari daerah A atau B menghasilkan 2 pola, maka struktur ini tidak dapat dihasilkan oleh algoritma ini. Dengan kondisi 3.2 disimpulkan bahwa daerah D adalah strip vertikal dan C strip horisontal. Satu dari dua pola yang dihasilkan daerah A (gambar 5b.) atau B (gambar 3.2c.) memiliki ukuran p_p dan l_p sehingga $2p_p \leq P$ dan $2l_p \leq L$. Gambar 3.2c. mengilustrasikan dengan jelas kondisi ini dengan kondisi 3.4 dimana untuk gambar 3.2c. didapat $1/4P < p_w$ dan $l_p \leq (P - 2p_w) \leq 1/2P$. Dengan menganggap strip dengan lebar l_p dengan Lemma 1, maka $k = 2$. Dan dengan strip yang sama dalam teori 1 dan kondisi 3.1 maka akan memberi nilai $V \geq 4/9 V^*$.

Untuk $k < 4$ algoritma selalu memberikan $V \geq 4/9 V^*$.

Untuk $k \geq 4$, dengan cara yang sama hanya ada 1 pola yang dapat masuk plat secara horisontal dan vertikal sebanyak 2 atau 3 kali yang memberikan solusi optimal ekivalen dengan strip horisontal atau vertikal.

Bukti melibatkan kondisi 3.1 dari Teori 2 untuk kasus pertama ($k = 2$ dan $\lfloor L/l_w \rfloor = 2$ atau Lemma 2 untuk kasus 2).



Gambar 3.2. Tiga pola pemotongan optimal (pola w memiliki panjang terkecil p_w yang dapat masuk ke dalam strip horizontal tepat 3 kali)

(a). Pola pemotongan yang dihasilkan algoritma.
 (b). Daerah A cukup untuk menghasilkan 2 pola, tidak dapat dengan algoritma ini.
 (c). Daerah B cukup untuk menghasilkan 2 pola dimana A hanya 1 pola, jadi tidak dapat dihasilkan algoritma ini.

Corollary 1:

Rasio aproksimasi ketika plat awal dan pola-pola yang akan dipotong adalah bujursangkar adalah $9/16$.

Bukti:

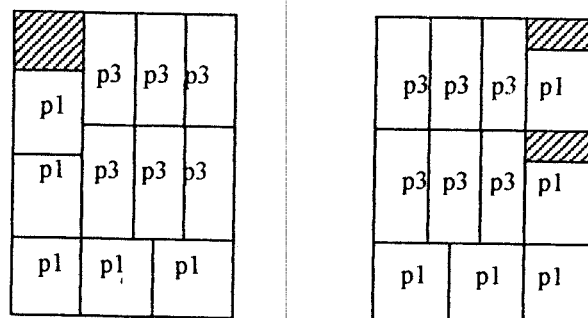
K sama seperti pada Teori 1. Kemudian rasio aproksimasi, diturunkan seperti pada teori 1, adalah $k^2 / (k+1)^2$. Untuk $k = 1, 2$ algoritma memberikan solusi dalam $1-\varepsilon$ untuk setiap $\varepsilon > 0$. $K = 3$ adalah worst case.

Contoh:

Jika beberapa batasan yang diberikan untuk ukuran pola-pola maka akan memberi hasil yang lebih meyakinkan. Sebagai contoh (seperti teori 1), bahwa jika ada sebuah pola (p_j, l_j) dengan $P = \rho p_j$ untuk beberapa nilai ρ dan $p_j \leq \lfloor P/k \rfloor$ untuk beberapa nilai k , maka diperoleh rasio aproksimasi $k/(k+1)$. Jika $k = 100$, maka akan diperoleh solusi-solusi optimal ($V > \frac{100}{101} V^*$).

3.2 ALGORITMA BEST STRIPS CUTTING (BSC)

Algoritma BSC (Best Strips Cutting Algorithm) disusun oleh beberapa strip vertikal dan (atau) horisontal pada plat awal atau sub-sub plat. Sebuah pola pemotongan optimal kerap kali strukturnya ekuivalen dengan yang lainnya. Gambar di bawah ini menggambarkan pola pemotongan optimal dengan tiga macam pola $p_1 = (3, 3)$, $p_2 = (4, 3)$, dan $p_3 = (2, 4)$ dipotong dari plat $(P, L) = (9, 11)$.



Gambar 3.3

- Pola pemotongan optimal, disusun oleh 2 sub plat $(3, 11)$ dan $(6, 11)$ masing-masing memiliki struktur strip. Dimensi untuk pola-pola $p_i = (P_i, L_i)$, $1 \leq i \leq 3$; 3×3 , 4×3 , 2×4 ; $V_i: (9, 12, 8)$
- Struktur strip yang ekuivalen.

Pada gambar 3.3(a) sub plat pertama memasukkan sebuah strip vertikal sedangkan yang kedua terdiri dari tiga strip horisontal. Dari metode ini diharapkan dapat menciptakan strip-strip optimal dan mengisinya ke plat awal atau beberapa sub plat untuk mendapatkan solusi yang memuaskan.

3.2.1 Identifikasi Masalah

Pemotongan bahan sering dilakukan dalam kehidupan sehari-hari. Bahan yang berbentuk lembaran persegi harus dipotong-potong menjadi bagian-bagian kecil agar sesuai dengan kebutuhan. Cara untuk menempatkan pola-pola yang

akan dipotong pada lembaran tadi akan menentukan apakah sisa bahan akan minimal atau tidak. Untuk mendapatkan profit yang optimal dari pemotongan bahan ini, maka penempatan pola-pola ini harus sangat diperhatikan.

Kemungkinan penempatan pola-pola tadi tidak hanya satu macam, tapi yang harus dipikirkan adalah bagaimana caranya mendapatkan penempatan pola-pola yang efisien sehingga sisa bahan minimal dan mendapatkan profit maksimal dari hasil yang didapatkan.

Masalah pemotongan benda dua dimensi yang akan diselesaikan dalam tugas akhir ini termasuk benda dua dimensi tak terbatas (*unconstrained*). Seperti pada penjelasan sebelumnya perbedaan antara variabel-variabel *bounded* dan *unbounded*, dimana variabel *bounded* menyatakan bahwa masing-masing pola yang akan muncul tidak boleh melebihi jumlah yang telah ditetapkan. Sedangkan variabel *unbounded* akan muncul beberapa kali dalam pola pemotongan.

Jika profit dihubungkan dengan masing-masing pola adalah luasan permukaannya, maka fungsi tujuan adalah meminimalkan sisa yang tidak terpakai. Sebaliknya fungsi tujuan untuk memaksimalkan fungsi profit dihubungkan dengan masing-masing penempatan pola.

Pemotongan bahan dua dimensi tak terbatas (*unbounded*) yang mempunyai sekumpulan pola-pola: $S = \{(p_1, l_1), (p_2, l_2), (p_3, l_3), \dots, (p_m, l_m)\}$. Inisialisasi ukuran plat yang akan dipotong untuk panjang P dan lebar L . Fungsi tujuan adalah untuk mencari pola pemotongan pada plat tanpa membatasi jumlah pengulangan masing-masing jenis pola dengan memaksimalkan fungsi profit:

$$F(P, L) = \sum_{i=1} a_i \pi_i \quad (3.3)$$

Dimana: a_i = banyaknya pola ke-i yang muncul

m = banyaknya macam pola dalam himpunan S

π_i = nilai profit pola ke-i = $(p \times l)$.

Jika jumlah pemunculan masing-masing pola tidak boleh melebihi angka tertentu maka disebut dengan masalah knapsack dua dimensi terbatas (*constrained*).

3.2.2 Metode Pemotongan Bahan Dua Dimensi

Ada beberapa metode pemotongan bahan dua dimensi. Sebelumnya telah dikemukakan sekilas, di sini akan dijelaskan lebih lanjut.

3.2.2.1 Orthogonal dan Non-Orthogonal

Jika tidak terdapat pembatas yang membatasi jumlah pemotongan yang dapat dibuat untuk menghasilkan potongan yang diminta dari bahan yang tersedia, tetapi hanya potongan tersebut harus paralel dengan tepi lembaran bahan maka pemotongan bersifat orthogonal.

Untuk memaksimalkan penggunaan bahan ada kemungkinan pemotongan dilakukan tidak orthogonal yaitu yang disebut non orthogonal. Kemungkinan ini untuk mengatasi potongan yang tidak optimal oleh orthogonal.

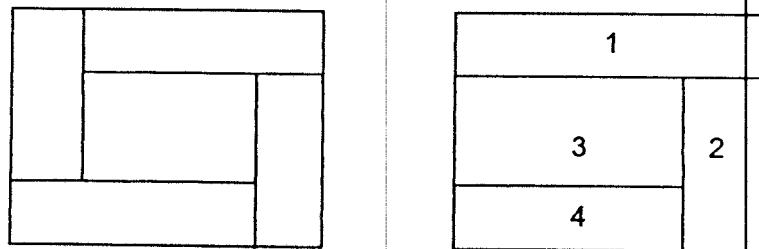
3.2.2.2 Guillotine dan non Guillotine

Untuk menghasilkan empat persegi panjang dari pemotongan bahan yang bersifat orthogonal maka lembaran bahan tadi harus dipotong dengan pemotongan guillotine. Pemotongan guillotine adalah cara memotong yang dimulai dari satu

sisi bahan tadi sampai ke sisi yang lainnya, sehingga akan dihasilkan 2 potongan empat persegi panjang baru yang ukurannya telah berkurang.

Asumsi-asumsi untuk menggambarkan keadaan sebenarnya yaitu:

1. Jenis alat pemotong adalah guillotine dimana pemotongan dilakukan secara orthogonal dari satu sisi ke satu sisi lain yang paralel dari plat atau dari beberapa sub-plat yang dihasilkan pemotongan sebelumnya.
2. Pemotongan dinormalisasi, yaitu sisa bahan yang dihasilkan dari pemotongan tidak pernah dipotong oleh algoritma dan berada pada plat bagian akhir yang akhirnya menghasilkan potongan sisa.
3. Semua pola berorientasi tetap (*fixed*), dimana ukuran akibat pemotongan yang dilakukan sangat tipis.



Gambar 3.4

- (a) Pola pemotongan yang tidak feasibel (bukan pemotongan guillotine)
 (b) Pola pemotongan yang feasibel dengan pemotongan guillotine

3.2.3 Trimming

Kondisi-kondisi trimming (sisa kecil) dari bahan setelah pemotongan dapat dinyatakan dalam 4 kondisi berikut ini:

1. Tidak boleh ada trimming, jika $P_i = x$, $L_i = y$, $i = 1, \dots, m$,
2. Trimming paralel ke x tapi tidak ada trimming paralel ke y , jika $P_i = x$, $L_i \leq y$, $i = 1, \dots, m$,

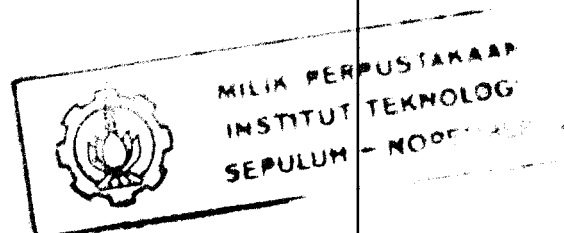
3. Trimming paralel ke y tapi tidak ada trimming paralel ke x, jika $P_i \leq x$,
 $L_i = y$, $i = 1, \dots, m$,
4. Trimming paralel ke x dan y jika $P_i \leq x$, $L_i \leq y$, $i = 1, \dots, m$.

3.2.4 Efek Simetris

Sebuah bahan berukuran (p, q) , maka akan ada $(p + q - 1)$ buah pemotongan yang dapat dibuat pada bahan sepanjang $x = 1, \dots, (p - 1)$, dan $y = 1, 2, \dots, (q - 1)$ dan 0-cut. Jika semua pemotongan ini dibuat, dihasilkanlah $p + q - 1$ buah sub plat, maka sub plat yang dihasilkan akan terduplikasi pada beberapa titik pemotongan. Hal ini dikarenakan munculnya pola pemotongan secara simetris. Untuk menghilangkannya dilakukan cara berikut ini.

Sebuah plat (p, q) , dipotong di titik $x = a$ menghasilkan dua sub plat berukuran (a, q) dan $(p - a, q)$. Jelasnya dua buah sub plat yang sama terjadi hasil dari pemotongan di $x = p - a$, yang secara simetris berseberangan pada $x = a$. Hal ini dapat dihindari tanpa kehilangan semua pola pemotongan yang unik, yaitu dengan membuat pemotongan hanya pada $1/2$ panjang bahan. Karena itu, range untuk pemotongan dibatasi $(p-1)$ sampai $1 < x < \lfloor p/2 \rfloor$ dimana $p/2$ ini berarti batas bawah yang tidak lebih dari pada $p/2$.

Hal serupa juga diterapkan untuk pemotongan ke arah sumbu y.



3.2.5 Efek Susunan Pemotongan

Bahan berukuran (p, q) dipotong pada $x = a$ menghasilkan dua buah sub plat (a, q) dan $(p - a, q)$. Pada sub plat berikutnya $(p - a, q)$ dipotong pada b , $a < b < [(p - a) / 2]$ yang menghasilkan tiga sub plat (a, q) , (b, q) , $(p - a - b, q)$ dari (p, q) .

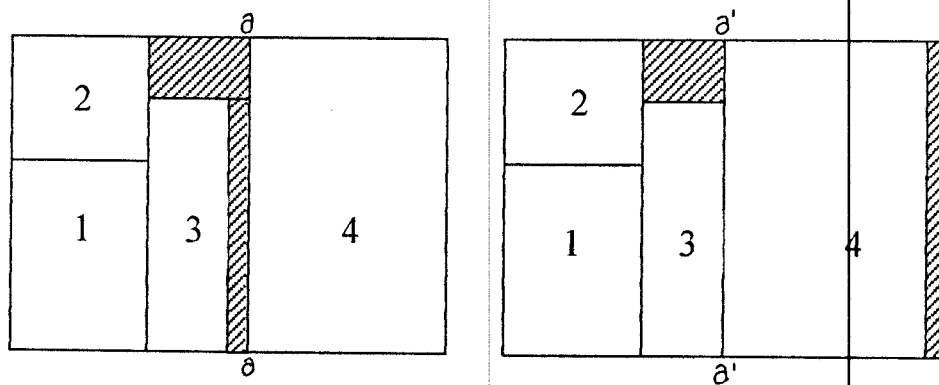
Tiga pola yang sama dapat juga dihasilkan oleh pemotongan pada $x = b$ dan baru kemudian pada $x = a$ di $(p - b, q)$. Jenis duplikasi ini dapat dihilangkan tanpa harus kehilangan pola pemotongan yang unik, yaitu dengan menyusun pemotongan secara bebas sehingga jika sebuah plat dipotong pada misalnya $x = \alpha$, kemudian semua sub plat x -cut pada resultan dua sub plat harus lebih besar dari α . Batasan ini dan batasan sebelumnya (yaitu efek simetris), mengharuskan bahwa untuk resultan dua buah sub plat yang lebih besar, (p, q) range dari x -cut akan dibatasi sampai $\alpha \leq x \leq [(p - \alpha) / 2]$ dan jika $(p - \alpha) / 2$ tidak dilakukan lagi x -cut. Untuk resultan dua sub plat lebih kecil dari batas tadi maka tidak ada x -cut yang mungkin dilakukan.

Konsekuensi dari susunan pola pemotongan sebagaimana dijelaskan di atas adalah untuk menghilangkan susunan yang berbeda dari pemotongan, ketika tercapai pola pemotongan akhir. Hal yang serupa juga diterapkan pada arah y .

3.2.6 Normalisasi Pemotongan

Jika bahan (p, q) dipotong pada posisi x , maka pola pemotongan akhir merupakan kombinasi dari panjang dari pola-pola yang tersedia untuk q , yang mana penjumlahannya harus tepat x . Jika tidak untuk arah x ditunjukkan $a - a'$ pada gambar 3.5a maka alternatif pemotongan pada posisi $x' < x$ dapat juga

menghasilkan pola pemotongan final (gambar 3.5b), yang melibatkan pola-pola yang sama dan karena memiliki nilai yang sama.



Gambar 3.5

- (a). Pola pemotongan non Normalisasi
- (b). Pola pemotongan yang di-Normalisasi

Gambar 3.5b disebut dengan pola normal. Ini menunjukkan setiap pola pemotongan mempunyai sebuah ekivalen normal. Normalisasi adalah sifat dari pola pemotongan yang berhubungan dengan sekumpulan pola-pola yang tersedia untuk dipotong yang mana sisa bahan tidak pernah dipotong oleh algoritma tapi hasil yang terjadi saat pemotongan pola pada bahan.

Dengan demikian (tanpa kehilangan optimalitas) seseorang dapat membatasi pemotongan arah x hanya untuk nilai-nilai dari x pola pemotongan-pola pemotongan normal dan demikian juga untuk arah y . Feasibel dari x pada bahan (p,q) memberikan himpunan S pola-pola untuk dipotong, harus pada nilai-nilai x yang merupakan elemen-elemen dari himpunan S tersebut.

3.2.7 Pembagian Plat Menjadi Sub-Plat

Berikut ini akan dijelaskan sub plat yang bagaimana yang dapat memenuhi strip-strip optimal dan bagaimana mengisinya secara efisien ke dalam sub plat.

Untuk mengisi plat dengan strip-strip secara efisien, terlebih dahulu dilakukan pemotongan terhadap plat menjadi sejumlah pasangan sub-plat. Masing-masing sub-plat ini diisi oleh strip-strip horisontal atau vertikal. Dengan cara ini struktur-struktur yang didapat merupakan kombinasi strip-strip horisontal-vertikal untuk memberikan solusi efisien.

Pembagian plat menjadi sub plat harus dilakukan hanya pada arah sumbu x-y yang memungkinkan algoritma memberikan solusi terbaik. Ada dua himpunan titik potong searah sumbu. Sebuah himpunan untuk arah horisontal yang ditandai P_P dan satu lagi untuk arah vertikal ditandai P_L . Himpunan-himpunan ini digunakan untuk mendefinisikan pola pemotongan guillotine yang dinormalisasi.

Normalisasi pola-pola guillotine:

$$P_P = \{x \mid x = \sum_{i=1}^m z_i p_i \leq \frac{1}{2}P, z_i \geq 0, z_i \text{ adalah integer}\}$$

$$P_L = \{x \mid x = \sum_{i=1}^m z_i l_i \leq \frac{1}{2}L, z_i \geq 0, z_i \text{ adalah integer}\}$$

Himpunan P_P dan P_L dibangkitkan dengan cara berikut:

$$g_i(x) = \begin{cases} 0 & \text{jika } i = 0 \wedge x = 0 \\ \infty & \text{jika } i = 0 \wedge x \neq 0 \\ g_{i-1}(x) & \text{jika } x < p_i, i \in I_i \\ \min\{g_{i-1}(x), \max\{1, \min_k \{g_{i-1}(x - kp_i), 1 \leq k \leq \lfloor x/p_i \rfloor, k \in N\}\}\}, \\ & i \in I, x \in X, \text{ di mana } I = \{1, 2, \dots, m\} \text{ dan } X = \{0, 1, 2, \dots, \lfloor 1/2P \rfloor\} \end{cases} \quad (3.4)$$

Hasil yang akan diperoleh titik x termasuk anggota himpunan P_P jika $g_m(x)=1$ dan tidak termasuk jika $g_m(x)=\infty$. Ditunjukkan kemudian bahwa hanya

elemen-elemen ini yang dapat meningkatkan hasil solusi. Pemotongan guillotine yang dilakukan pada $x \in (P_p \cup P_L)$ akan meningkatkan hasil.

3.2.8 Pembentukan Strip

Disini akan dijelaskan prosedur untuk membangkitkan strip horisontal dan vertikal dengan memenuhi panjang dan lebar yang diberikan. Dimisalkan panjang = α sub plat dan lebar = β sub plat.

3.2.8.1 Strip-Strip Horisontal

$S = \{(p_1, l_1), (p_2, l_2), (p_3, l_3), \dots, (p_m, l_m)\}$ adalah himpunan pola yang akan dipotong. Pola ini disusun dalam urutan lebarnya yang semakin naik. Untuk menyatakan banyaknya lebar yang berbeda dinyatakan dalam r . Indeks terlebar h_i dinyatakan dengan k_i , $i = 1..r$.

$S_{k_i} = \{p_j: (p_j, l_j) \in S, l_j \leq l_{k_i}\}$ adalah himpunan anggotanya semua pola dengan lebar yang lebih kecil atau sama dengan pola yang memiliki lebar terbesar. Sehingga didapat $S_{k_1} \subset S_{k_2} \subset S_{k_i}$. Untuk masing-masing S_{k_i} diasosiasikan sebagai knapsack satu dimensi ODK(i) dengan fungsi profitnya:

$$\text{Max: } \sum_{l_\mu \in S_{k_i}} \pi_\mu x_\mu \quad (3.5)$$

$$\text{Syarat: } \sum_{l_\mu \in S_{k_i}} p_\mu x_\mu \leq \alpha, \quad x_\mu \geq 0, \quad x_\mu \text{ integer}$$

Dimana:

π_μ = profit untuk pola (p_μ, l_μ)

x_μ = banyaknya pola (p_μ, l_μ) pada strip ke i .

Dengan menyelesaikan ODK(i), $i = 1, 2, \dots, r$ dibuatkan strip-strip horisontal dengan panjang α lebar h_{ki} dan nilai profit $F(\alpha)$. Untuk sederhananya nilai h_{ki} kemudian diganti dengan β .

3.2.8.2 Strip-Strip Vertikal

Dengan mengulangi cara yang sama seperti yang telah dilakukan pada strip horisontal, akan dibangkitkan strip-strip vertikal untuk sub plat-sub plat. Pola ini disusun dalam urutan panjang yang makin besar. Untuk menyatakan banyaknya panjang yang berbeda dinyatakan dengan v . Indeks terpanjang p_i dinyatakan dengan k_i , $i = 1..v$.

$S_{ki} = \{l_j: (p_j, l_j) \in S, p_j \leq p_{k_i}\}$ adalah himpunan anggotanya semua pola dengan panjang yang lebih kecil atau sama dengan pola yang memiliki panjang terbaik.

Strip-strip vertikal ini didapatkan dengan cara yang sama dengan strip-strip horisontal.

3.2.9 Penempatan Strip Pada Bahan.

Setelah strip-strip horisontal dan vertikal dibangkitkan maka dilanjutkan dengan menempatkannya ke dalam plat. Masing-masing strip ini memiliki nilai profit $F_i(\alpha)$ dan tingginya β_i (untuk strip-strip horisontal) dan $F_i(\beta)$ dan panjangnya α_i (untuk strip-strip vertikal).

Strip-strip yang terbaik (nilai profit terbesar) akan ditempatkan ke dalam plat dengan memperhitungkan lebar untuk strip-strip horisontal (atau panjang untuk strip-strip vertikal). Penempatan ini merupakan penyelesaian masalah

knapsack satu dimensi yang berikutnya. Untuk strip-strip horisontal, knapsack didefinisikan sebagai berikut:

$$\begin{aligned} & \text{ODK}(r+1) \\ & F_{r+1}(\beta) = \max \sum_{i=1} F_i(\alpha) y_i \\ & \text{Syarat: } \sum_{i=1} \beta_i(\alpha) y_i \leq \beta, y_i \geq 0, y_i \text{ integer} \end{aligned} \quad (3.6)$$

Dimana:

β = lebar plat yang akan dipotong,

$y_i, i = 1, \dots, r$ menyatakan jumlah pengulangan strip horisontal ke-i.

Untuk mengisi plat dengan strip-strip vertikal seperti yang didefinisikan ODK(v+1) dapat dilakukan dengan cara yang sama yaitu menggantikan simbol yang digunakan.

$$\begin{aligned} & \text{ODK}(v+1) \\ & F_{v+1}(\alpha) = \max \sum_{i=1} F_i(\beta) y_i \\ & \text{Syarat: } \sum_{i=1} \alpha_i(\beta) y_i \leq \alpha, y_i \geq 0, y_i \text{ integer} \end{aligned} \quad (3.7)$$

Dimana:

α = panjang plat yang akan dipotong,

$y_i, i = 1, \dots, v$ menyatakan jumlah pengulangan strip vertikal ke-i.

Setelah penempatan selesai dilakukan maka terbentuklah kombinasi pola potong yang disebut pola pemotongan. Pola pemotongan ini yang dihasilkan dari kombinasi strip-strip vertikal dan horisontal memiliki nilai-nilai sebagai berikut:

$$V^{\text{hor}} = F_{r+1}(\beta) \text{ dan } V^{\text{ver}} = F_{v+1}(\alpha)$$

Prosedur yang digunakan membutuhkan $r + v + 2$ ($r, v < m$) buah knapsack yang digunakan untuk menghasilkan strip-strip, sehingga semua anggota S dapat masuk ke dalam plat (sub plat). Jumlah knapsack ini dapat berkurang dengan menggunakan metode program dinamis.

(α, β) adalah inisial dari plat P dan L . Penyelesaian terhadap knapsack terbesar r , menghasilkan strip horisontal terlebar. Kemudian diikuti oleh penyelesaian terhadap knapsack yang lainnya $1, 2, \dots, r-1$ dan juga oleh semua strip dari lebar yang tersedia. Sebenarnya hanyalah satu knapsack yang ditempati oleh semua strip-strip horisontal itu. Berikutnya semua nilai, yang dibutuhkan oleh prosedur pengisian untuk merumuskan knapsack ke $-(r + 1)$, dicari untuk memilih strip-strip yang akan mengisi plat.

Hal yang sama, knapsack ke v membuat strip vertikal terpanjang dan juga semua strip-strip dengan panjang yang tersedia.

Jadi akhirnya $r + v + 2$ buah knapsack yang dibutuhkan untuk mengisi plat secara horisontal dan vertikal akan berkurang sampai 4 buah knapsack.

Semua sub strip optimal yang dibangkitkan untuk sub plat-sub plat setelah pembagian pada titik-titik x dalam himpunan P_P dan P_L telah tersedia. Untuk masing-masing sub plat yang diselesaikan adalah dua knapsack yang akan memilih sub strip terbaik untuk mengisi sub plat tersebut.

Dari penjelasan di atas, akan disimpulkan dalam langkah-langkah utama yang dilakukan dalam algoritma aproksimasi (yang dinyatakan dalam algoritma BSC) untuk menyelesaikan masalah knapsack dua dimensi.

Langkah-Langkah Utama Dari Algoritma BSC

Langkah 1: Menentukan panjang ($1/2 P$) dan lebar ($1/2 L$) plat awal menjadi dua buah himpunan P_P dan P_L . Dimana P_P ini adalah himpunan kombinasi linear koefisien integer dari panjang dan P_L adalah himpunan kombinasi dari koefisien integer dari lebar pola-pola yang akan dipotong.

$$P_P = \{x \mid x = \max \sum_{i=1}^m z_i p_i \leq \frac{1}{2} P, z_i \text{ integer positif}\},$$

$$P_L = \{y \mid y = \max \sum_{i=1}^m z_i l_i \leq \frac{1}{2} L, z_i \text{ integer positif}\},$$

untuk mendapatkan x dan y dibangkitkan dengan persamaan berikut ini;

$$g_i(x) = \begin{cases} 0 & \text{jika } i = 0 \wedge x = 0 \\ \infty & \text{jika } i = 0 \wedge x \neq 0 \\ g_{i-1}(x) & \text{jika } x < p_i, i \in I \\ \min\{g_{i-1}(x), \max\{1, \min_k \{g_{i-1}(x - kp_i), 1 \leq k \leq \lfloor x/p_i \rfloor, k \in N\}\}\} & \text{jika } i \in I, x \in X, \text{ di mana } I = \{1, 2, \dots, m\} \text{ dan } X = \{0, 1, 2, \dots, \lfloor 1/2 P \rfloor\} \end{cases}$$

Jadi titik x adalah anggota P_P jika $g_m(x) = 1$ dan x bukan anggotanya jika $g_m(x) = \infty$.

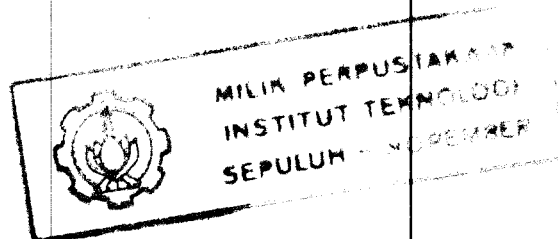
Langkah 2: Membangkitkan strip horisontal tertinggi dan strip vertikal yang terpanjang.

$S_{ki} = \{p_j : (p_j, l_j)S, l_j \leq l_{ki}\}$ untuk strip horisontal, sedangkan

$S_{ki} = \{l_j : (p_j, l_j)S, p_j \leq p_{ki}\}$ untuk strip vertikal.

Langkah 3: Pada masing-masing elemen $x_j \in (P_P \cup P_L)$ akan dilakukan pemotongan guillotine vertikal jika $x_j \in P_P$ atau pemotongan guillotine horisontal jika $x_j \in P_L$. Sehingga dari pemotongan ini akan dihasilkan 2 buah sub plat.

Diasumsikan: $P_P = (x_j, L)$ dan $P_L = (P - x_j, L)$, adalah sub-sub plat yang dihasilkan dari pemotongan di titik $x \in P_P$, sedangkan $P_L = (P, x_j)$ dan $P_L = (P, L - x_j)$, dihasilkan dari pemotongan di titik $x \in P_L$.



Langkah 3.1: Menempatkan strip-strip horisontal ke dalam sub plat yang telah ada P_i , $i = 1, 2$.

Nilai V_{hor_i} menyatakan nilai-nilai profit yang berhubungan dengan pola pemotongan-pola pemotongan yang dihasilkan. Nilai profit untuk masing-masing strip tersebut dapat dihitung:

$$ODK(i) = F_i(\alpha) = \text{Maks} \sum_{p_\mu \in S_{ki}} \pi_\mu p_\mu,$$

syarat yang harus dipenuhi: $\sum_{p_\mu \in S_{ki}} \pi_\mu p_\mu \leq \alpha, x_\mu \geq 0, x_\mu$ adalah integer.

Langkah 3.2: Step 3.1 diulangi untuk menempatkan strip-strip vertikal pada sub plat P_i , $i = 1, 2$.

Nilai V_{ver_i} menyatakan nilai-nilai profit pola pemotongan baru, yang dihasilkan dari:

$$ODK(i) = F_i(\beta) = \text{Maks} \sum_{p_\mu \in S_{ki}} \pi_\mu p_\mu,$$

syarat yang harus dipenuhi: $\sum_{p_\mu \in S_{ki}} \pi_\mu p_\mu \leq \beta, x_\mu \geq 0, x_\mu$ adalah integer.

Langkah 3.3: Karena $V_i = \text{maks}(V_{hor_i}, V_{ver_i})$ menyatakan nilai profit pola pemotongan yang maksimal, maka $V_1 = \text{maks}(V_{hor_1}, V_{ver_1})$ dan $V_2 = \text{maks}(V_{hor_2}, V_{ver_2})$ Simpan pola pemotongan terbaik untuk masing-masing sub plat P_i , $i = 1, 2$.

$$ODK(r+1) = F_{r+1}(\beta) = \text{Maks} \sum_{i=1}^r F_i(\alpha) y_i$$

Dimana syarat yang harus dipenuhi: $\sum_{i=1}^r \alpha_i y_i \leq \beta, y_i \geq 0, y_u$ adalah integer.

$$ODK(v+1) = F_{v+1}(\alpha) = \text{Maks} \sum_{i=1}^r F_i(\beta) y_i,$$

Dimana syarat yang harus dipenuhi: $\sum_{i=1}^v \beta_i y_i \leq \alpha, y_i \geq 0, y_u$ adalah integer.

Langkah 3.4: Simpan pola pemotongan yang dihasilkan untuk plat dan nilainya nyatakan dalam

$$V^{xj} = (V_1 + V_2)$$

Langkah 4: Dari semua titik $x_j \in (P_L \cup P_H)$, dipilih pola pemotongan yang memiliki nilai V^{xj} maksimum

3.2.10 Fase 0-Cut

Fase 0-cut terjadi jika membagi plat dengan elemen 0 dari P_P , sub plat kedua adalah plat awal dengan dimensi (P, L) . Fase ini melakukan langkah ke 2 dan 3 dari algoritma di atas dan $X_j = 0$. Hasil yang didapatkan $V^0 = \max(V^{hor}_2, V^{ver}_2)$ dengan menyelesaikan hanya 4 buah knapsack.

Pada saat melakukan normalisasi pemotongan guillotine, dengan hanya memakai elemen P_P dan P_L untuk melakukan pemotongan guillotine pada plat awal, jelas komputasi akan berkurang. Hal ini dibandingkan jika dilakukan pada tiap titik selain titik-titik normalisasi sepanjang plat.

Seperti yang telah dijelaskan sebelumnya bahwa pada saat fase 0_cut, algoritma hanya membutuhkan 4 buah knapsack saja. Dan pada masing-masing sub plat yang dibangkitkan yang dibutuhkan adalah 2 buah knapsack. Oleh karena itu BSC mengeksekusi semua elemen dari $(P_P \cup P_L)$ membutuhkan untuk masing-masing elemen adalah 4 buah knapsack. Sehingga total jumlah knapsack yang dibutuhkan BSC menjadi 4 $(P_P \cup P_L)$.

Knapsack terbesar yang ditempati adalah pada penyelesaian di langkah kedua. Yang pertama dengan ukuran m (jumlah item yang akan dipotong) dan ukuran keduanya P , sedangkan yang kedua dengan ukuran m dan ukuran

keduanya L . Dengan menggunakan program dinamis, worst case yang diperoleh adalah $O(m \cdot \max(P, L))$ dalam masalah knapsack ini ukuran tidak polinomial.

```

Procedure Knapsack (alpha, betha, p, l,  $\pi$ )
for i  $\leftarrow$  1 to m
  do for y  $\leftarrow$  alpha downto 1
    do MaxCurrent  $\leftarrow$  0
    Batas  $\leftarrow$  y / W[i]
    if Batas  $\geq$  1 then
      for k  $\leftarrow$  1 to Batas
        do x  $\leftarrow$  (y - k * W[i])
        if x  $\geq$  0 then
          MaxTemp  $\leftarrow$  Tabel[i-1, x] + k * V[i]
          if MaxCurrent < MaxTemp then
            MaxCurrent  $\leftarrow$  MaxTemp
            Frekw  $\leftarrow$  k
          if MaxCurrent  $\geq$  Tabel[i, y] then
            Tabel[i, y]  $\leftarrow$  MaxCurrent
            TabelIndex[i, y]  $\leftarrow$  i
          Else Tabel[i, y]  $\leftarrow$  Tabel[i-1, y]
            TabelIndex[i, y]  $\leftarrow$  0
        Tabel[i, y]  $\leftarrow$  Tabel[i-1, y]
        TabelIndex[i, y]  $\leftarrow$  0

```

Pada evaluasi terhadap *running time* untuk prosedur di atas didapatkan $O(n^3)$, yang merupakan kondisi *worst case*-nya. Selain *running time* untuk prosedur di atas juga diperhitungkan untuk prosedur sorting, yang mana bila m yang dibutuhkan cukup besar maka pemilihan jenis sorting akan menentukan sekali. Pada BSC ini prosedur sorting yang diterapkan adalah *bubble sort* yang memberikan nilai *worst case* sebesar $O(n^2)$.

BAB IV

PERANCANGAN DAN PEMBUATAN

PERANGKAT LUNAK

Perancangan perangkat lunak ini merupakan implementasi dari langkah-langkah utama yang telah diuraikan dalam bab sebelumnya. Pembahasan selanjutnya melingkupi gambaran umum perangkat lunak itu yang akan diperjelas dengan diagram aliran data, perancangan data (masukan, proses serta keluaran) dan perancangan menu.

Penyusun utama perangkat lunak yang dinamakan dengan *Best Strips Cutting* ini, terdiri dari dua bagian utama yaitu proses penyiapan data-data untuk diproses dan proses knapsack itu sendiri. Perangkat lunak ini dirancang untuk mengimplementasikan algoritma aproksimasi guna mendapatkan pola pemotongan dengan solusi yang optimal. Perancangan ini diimplementasikan menggunakan compiler Borland Delphi under windows.

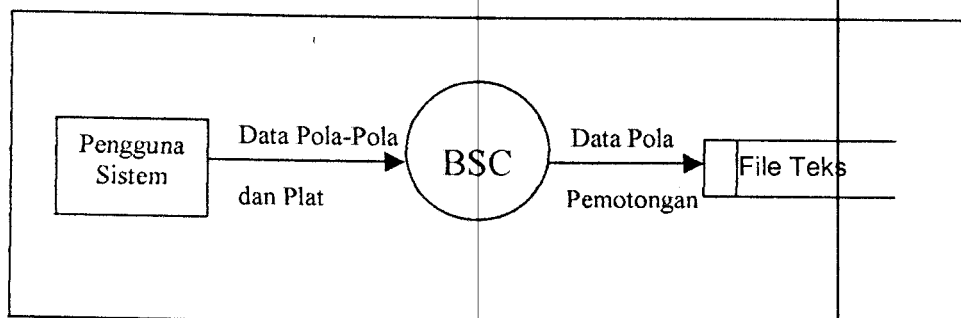
4.1 GAMBARAN UMUM PERANGKAT LUNAK

Dalam tugas akhir ini dirancang suatu perangkat lunak untuk mendapatkan pola pemotongan optimal dalam pemotongan bahan dua dimensi dengan algoritma aproksimasi. Secara umum dapat dijelaskan gambaran perangkat lunak melalui

keterkaitan antara data masukan, proses dan keluaran. Masukan dari pengguna sistem berupa data plat dan pola-pola. Data masukan ini dapat dimasukkan langsung dan dapat juga melalui file teks. Proses pencarian kombinasi pola pemotongan akan dilakukan setelah data masukan dianggap benar. Aliran data dalam sistem ini akan dijelaskan dengan Diagram Aliran Data (DAD).

4.2 DIAGRAM ALIR DATA (DAD)

Diagram alir data untuk perangkat lunak ini ditunjukkan pada gambar-gambar berikut ini. Gambar 4.1 adalah diagram alir data level-0 yang menggambarkan sistem pencarian pola pemotongan secara umum. Sistem ini menerima data masukan dari user berupa data-data pola dan plat yang akan dipotong.



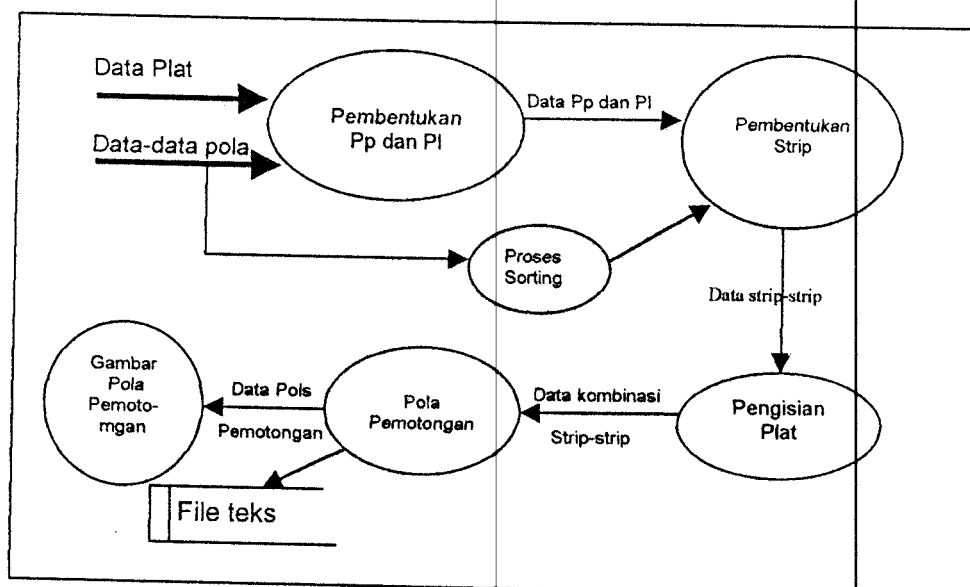
Gambar 4.1 DAD level 0, Sistem perangkat lunak

Proses utama diagram ini adalah Best Strips Cutting yang akan dipecah menjadi beberapa proses dan sub proses. Dan hasil akhir dari proses ini adalah pola pemotongan yang merupakan kombinasi pola-pola dalam strip-strip.

Pada gambar 4.2 diagram alir data level-1 ini, proses utama tadi telah dipecah menjadi proses-proses pembentukan P_p dan P_L , pembentukan strip-strip,

pengisian plate dan pencarian pola pemotongan. Pada proses pembentukan strip-strip terdapat sub proses sorting data pola yaitu terhadap panjang dan lebar pola.

Secara umum proses utama meliputi proses pembentukan strip-strip dan pengisiannya ke dalam plat. Data masukan pada proses utama adalah data plat dan data pola-pola yang dianggap sebagai permasalahan yang akan diselesaikan. Dari proses pembentukan P_p dan P_L menghasilkan data titik potong untuk pemotongan guillotine. Data ini menjadi masukan untuk proses pembentukan strip-strip bersama dengan data pola yang telah diurut pada proses sorting. Data ini diproses untuk menjadi strip-strip optimal yang kemudian menjadi data masukan pada proses pengisian plat. Proses ini menghasilkan kombinasi-kombinasi pola pemotongan feasibel. Untuk mendapatkan kombinasi optimal di antara kombinasi feasibel ini maka akan dipilih yang distribusi sisanya yang minimal.



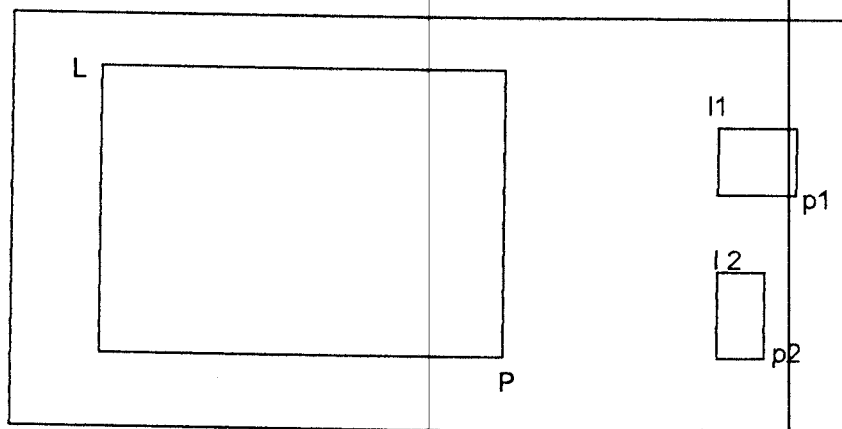
Gambar 4.2 DAD level 1, Proses pencarian pola pemotongan

4.3 PERANCANGAN DATA

Telah dijelaskan sebelumnya bahwa masukan yang dibutuhkan perangkat lunak ini adalah data plat dan data pola untuk masukan, proses dan keluaran. Untuk data masukan ini yang dimasukkan adalah ukuran panjang dan lebar dari plat maupun pola-pola.

4.3.1 Data Masukan

Data masukan plat ini terdiri satu data saja yaitu panjang (P) dan lebar (L), dengan ukuran plat maksimum dibatasi sampai 1000 satuan. Sedangkan data pola yang dibutuhkan yaitu panjang (p) dan lebar (l) yang lebih dari satu buah data. Karena data pola tidak kurang dari 1 maka akan dibuatkan array untuk menampung semua data tersebut. Banyaknya pola (m) dibatasi sampai 50 buah data.



Gambar 4.3 Model plat (P, L) , pola (p1, l1) dan (p2, l2)

Struktur untuk data masukan ini sebagai berikut:

```
SzPola = Record
    Pola : Array[1..50, 1..2] of integer;
    P, L : integer;
    nPola : integer;
End;
```

Data yang dimasukkan dapat disimpan dalam file teks < *namafile.txt* > yang nantinya dapat di-update kembali.

Dari data yang dimasukkan di atas kemudian diubah menjadi data siap untuk diproses. Data plat (P, L) diubah menjadi kapasitas dari masalah knapsack tadi. P adalah kapasitas untuk penempatan pola dalam arah horisontal dan L adalah kapasitas untuk arah vertikal.

Untuk data pola, panjang pola (p) menjadi item yang akan dimasukkan ke dalam knapsack dengan kapasitasnya P. Hal yang sama juga dilakukan untuk lebar pola (l) dengan kapasitas L.

4.3.2 Data Saat Pemrosesan

Proses-proses yang akan dilakukan *Best Strips Cutting* adalah Pembentukan P_P dan P_L , Pembentukan Strip, Pengisian Plat dan mencari Pola pemotongan. Dari masing-masing proses ini memerlukan data-data sebagai berikut:

1. Pembentukan P_P dan P_L ,

Data masukan yang telah dijelaskan di atas dipakai dalam proses untuk menghasilkan titik-titik potong (x_j). Data P_P dan P_L ini disimpan dalam array satu dimensi.

2. Pembentukan Strip,

- Dari hasil proses P_P dan P_L yaitu titik-titik x_j , dilakukan pemotongan dengan pemotongan guillotine. Dari tiap titik pemotongan ini didapatkan dua buah

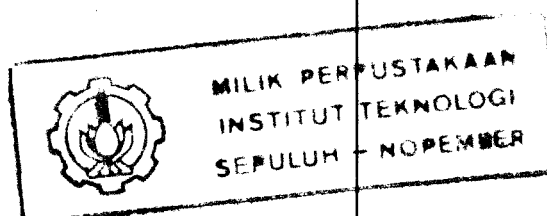
knapsack dengan panjang dan lebar yang baru. Ukuran panjang dan lebar ini menjadi ukuran kapasitas yang baru.

- Data masukan dan kapasitas baru hasil proses di atas ini diperlukan dalam proses untuk menghasilkan keluaran berupa strip-strip pada proses pembentukan strip. Strip-strip ini didefinisikan sebagai ODK-ODK yang dibutuhkan oleh proses berikutnya. Strip ini merupakan kombinasi pola optimal yang terpilih.
- Untuk arah horisontal dengan menggunakan data pola (p) dimasukkan ke dalam dua buah knapsack sehingga dihasilkan strip horisontal-1 dan strip horisontal-2.
- Hal yang sama juga untuk arah vertikal dengan data pola (l) sehingga didapatkan strip vertikal-1 dan strip vertikal-2.

Data strip-strip ini disimpan dalam tabel array dua dimensi yaitu TabelIndeks.

3. Pengisian Plat,

- Data yang dipakai adalah keluaran dari proses pembentukan strip dan data pola awal. Strip-strip tersebut akan ditempatkan ke dalam knapsack dalam arah dimensi kedua (yaitu kapasitas yang diambil dari lebar plat tersebut).
- Proses yang dilakukan di sini adalah mencari kombinasi pola-pola feasibel yang akan diisikan ke dalam plat. Kombinasi ini termuat dalam masing-masing knapsack sebanyak $n(P_P \cup P_L)$. Untuk mengurangi jumlah knapsack



ini digunakan program dinamis. Data hasil proses ini disimpan dalam array dua dimensi.

4. Pola pemotongan

Dari data hasil proses pengisian plat di atas akan dicari solusi yang paling optimal dengan sisa minimal yang akan menjadi pola pemotongan dari plat awal.

4.3.3 Data Keluaran

Keluaran dari BSC ini berupa kombinasi strip-strip yang disusun secara vertikal atau horisontal. Strip-strip optimal ini merupakan kombinasi pola-pola terpilih sebelumnya.

Keluaran ini akan ditampilkan sebagai pola pemotongan yang akan digunakan sebagai instruksi pemotongan untuk plat yang diberikan. Solusi dari BSC ini yang dapat ditampilkan selain pola pemotongan yang optimal juga solusi-solusi yang didapatkan dari hasil normalisasi. Jadi jumlah solusi yang dapat ditampilkan

$$\sum \text{Solusi Feasibel} = \sum n(P_P \cup P_L).$$

Visualisasi dari solusi-solusi feasibel yang didapatkan dari normalisasi ini ditampilkan di samping hasil pola pemotongan (solusi optimal) itu sendiri.

Keluaran dari proses pola pemotongan ini juga disimpan dalam file teks <namafile.txt>. Informasi yang disimpan dalam file ini adalah kombinasi pola-pola yang menyusun strip-strip tersebut yang memberi solusi optimal.

4.4 PERANCANGAN PROSES

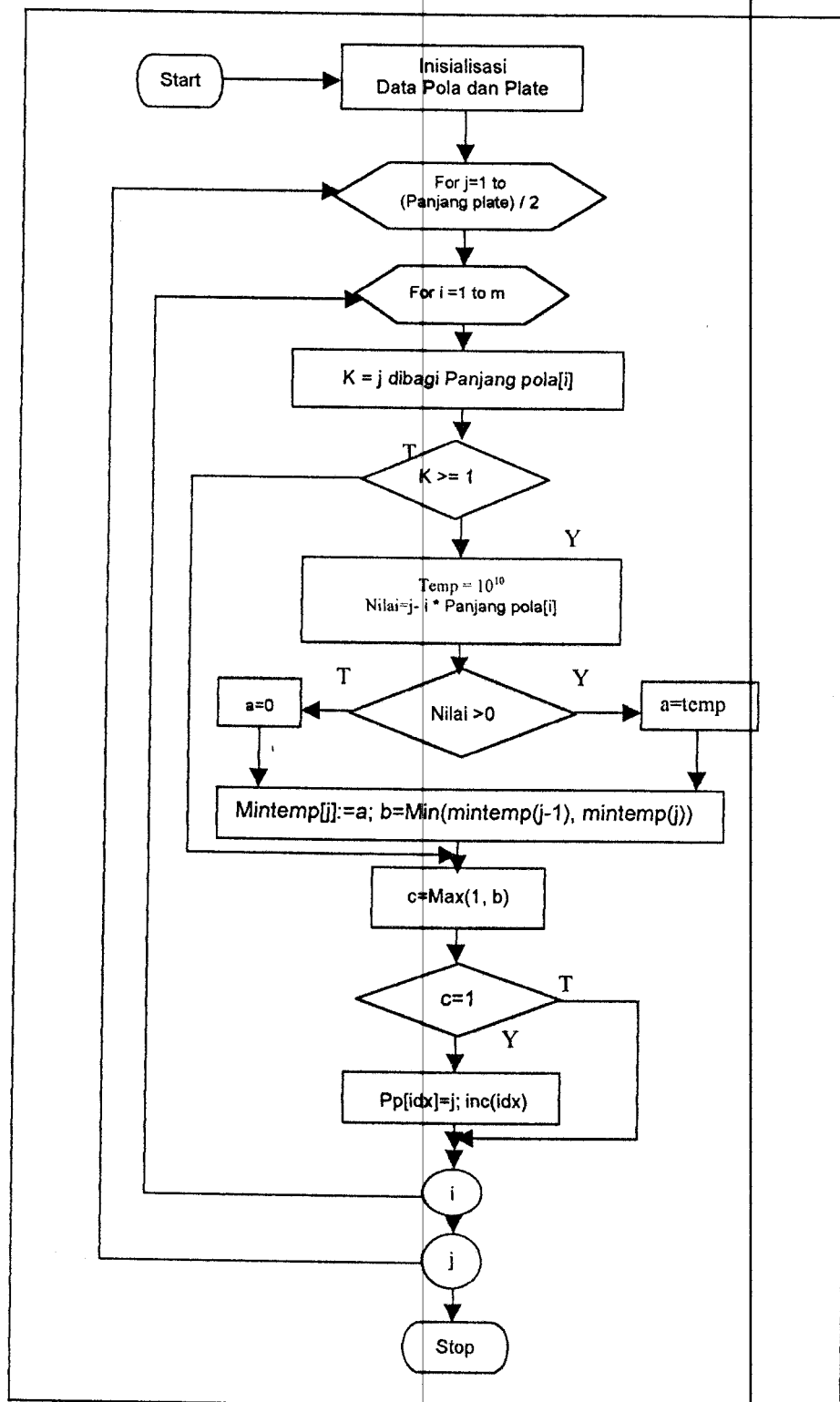
Proses-proses utama dari sistem ini terdiri dari proses pembentukan P_P dan P_L , proses pembentukan strip-strip, proses pengisian plat dan proses pencarian pola pemotongan. Untuk masing-masing proses akan diperjelas dengan diagram alir.

4.4.1 Pembentukan P_P dan P_L

Langkah-langkah untuk melakukan pembentukan P_P dan P_L seperti tergambar pada gambar 4.5 yaitu diagram alir proses ini.

Dalam proses untuk menghasilkan titik-titik x anggota dari P_P , titik-titik ini harus $\leq P/2$. Batas baru ini dibagi dengan panjang pola ke i . Jika $k \geq 1$, dicari nilai $g_{i-1}(x-k.p_i)$ yang minimum. Kemudian nilai minimum dari $g_{i-1}(x-k.p_i)$ dibandingkan dengan 1 untuk mendapatkan nilai maksimum. Yang terakhir nilai maksimum ini dibandingkan kembali dengan $g_{i-1}(x)$ dan diambil nilai yang minimum. Jika nilai akhir = 1 maka titik x termasuk anggota P_P . Dan jika nilai = ∞ , maka titik x bukan termasuk anggota P_P .

Untuk menghasilkan x anggota P_L hal yang sama dilakukan dimana batas yang digunakan adalah $L/2$. Titik-titik x ini akan menjadi titik dimana dilakukan pemotongan guillotine. Untuk x anggota P_P adalah pemotongan guillotine arah vertikal sedangkan untuk x anggota P_L adalah pemotongan guillotine arah horisontal.



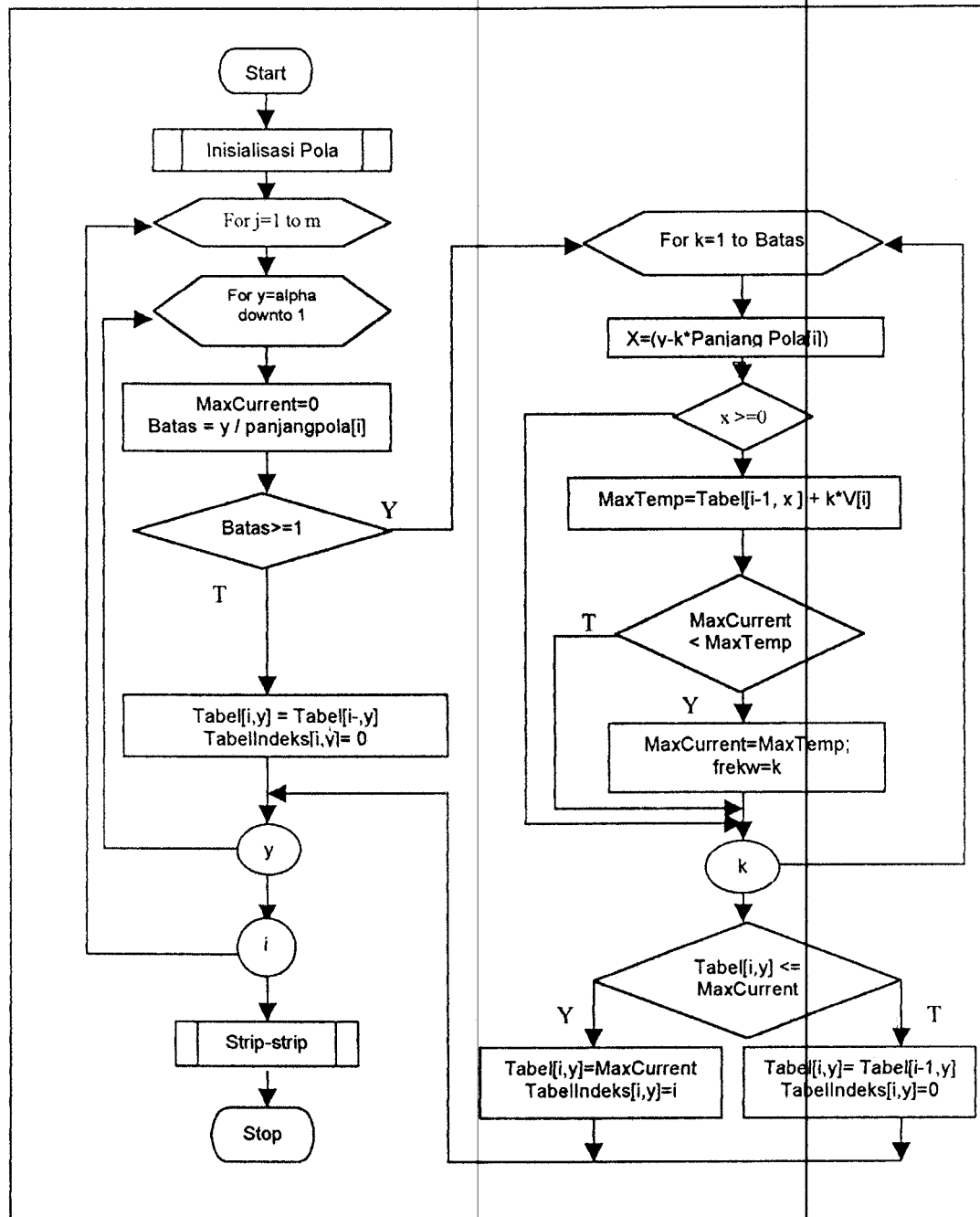
Gambar 4.4 Diagram alir pembentukan P_p dan P_L

4.4.2 Pembentukan Strip

Sebelum dilakukan proses pembentukan strip dilakukan sorting terhadap data pola. Untuk proses pembentukan strip horisontal data lebar yang disorting sedangkan untuk strip vertikal maka yang disorting adalah data panjang. Kemudian x yang menjadi titik untuk pemotongan guillotine akan menjadikan plat menjadi dua buah knapsack baru. Masing-masing ukuran knapsack baru ini dijadikan nilai kapasitas. Untuk strip horisontal, kapasitas dinyatakan dengan α yaitu panjang knapsack dan untuk vertikal dengan β yaitu lebar knapsack. Luas pola merupakan nilai yang didefinisikan untuk masing-masing pola tersebut.

Jika data awal terbentuk maka mulai dilakukan pencarian kombinasi pola untuk mendapatkan strip-strip. Strip-strip yang pertama dicari adalah mencari strip-strip horisontal. Jumlah pola yang dapat dimasukkan ke dalam kapasitas yang tersedia dihitung dengan membagi α dengan panjang pola ke i ($\text{batas} = \alpha/p_i$). Jika batas ini ≥ 1 maka hitung sisa kapasitas setelah ditempati sebuah pola ke i ini. Jika masih bersisa pola tadi dimasukkan kembali sampai kapasitas = 0. Kemudian menghitung profit dari pemasukan pola tadi yang menjadi nilai sementara dari tabel $[i, y]$. Profit ini dibandingkan dengan profit sebelumnya jika lebih besar maka nilai sekarang adalah profit yang baru dihitung. Jika tidak, maka nilai adalah profit sebelumnya yaitu nilai tabel $[i-1, y]$.

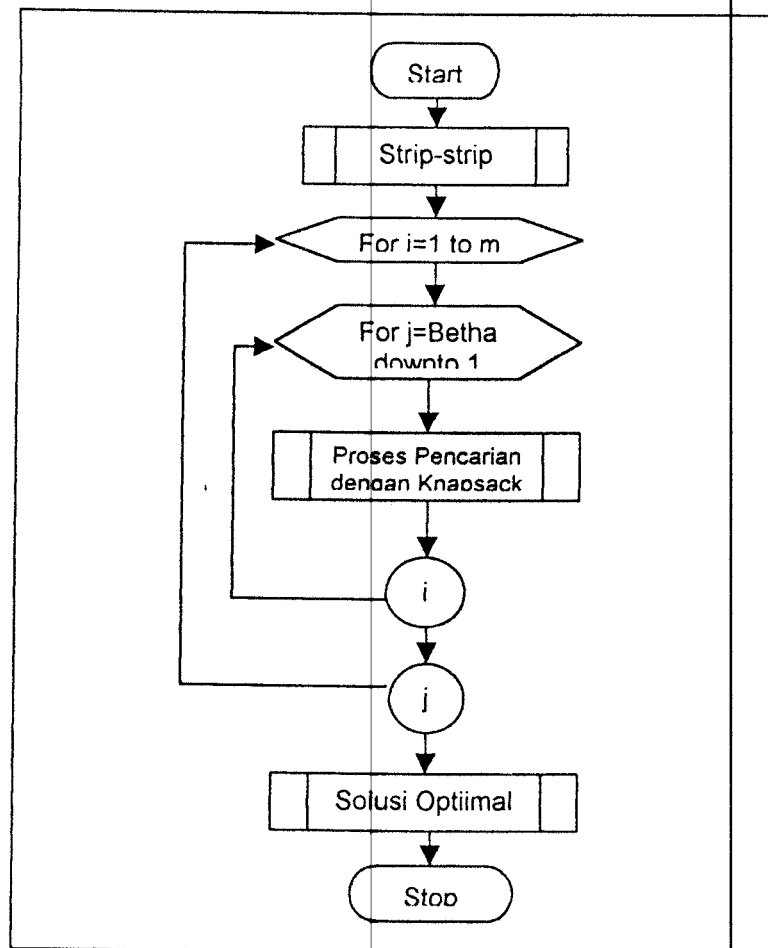
Selanjutnya proses berulang untuk pola selanjutnya. Hasil akhir dari proses ini didapat strip-strip horisontal sebanyak m buah. Strip-strip ini kemudian akan ditempatkan ke plat proses pengisian plat.



Gambar 4.5 Diagram alir pembentukan strip

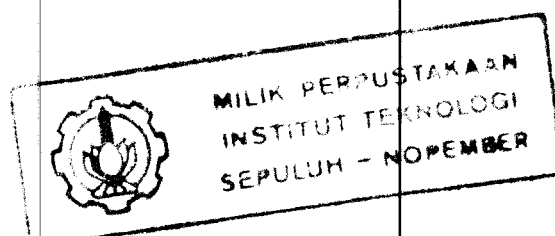
4.4.3 Pengisian Plat

Langkah-langkah dari proses pengisian plat yang dilakukan disini pada prinsipnya sama dengan langkah-langkah dalam proses pembentukan strip-strip. Ukuran lebar plat (L) menjadi kapasitas dari knapsack. Profit dari strip-strip menjadi nilai dari pola-pola tersebut.



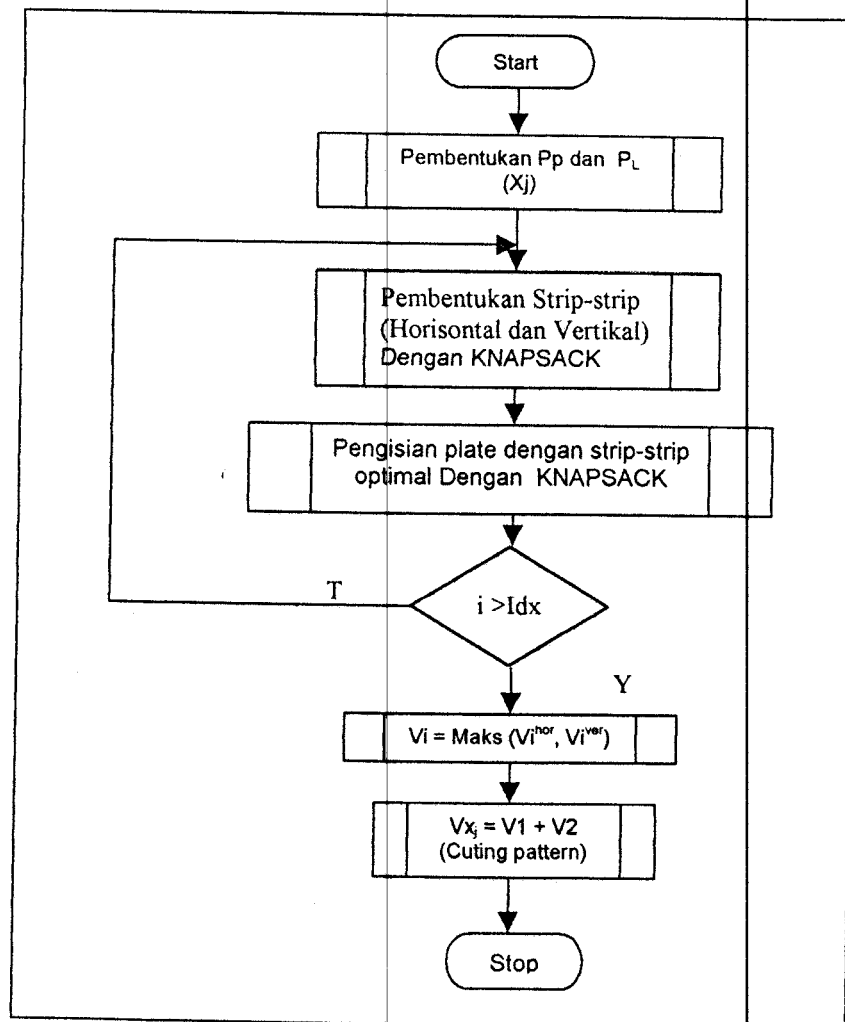
Gambar 4.6 Diagram alir pengisian plat

Keluaran dari proses ini adalah kombinasi dari strip-strip yang merupakan solusi feasibel untuk knapsack hasil pemotongan guillotine.



4.4.4 Pola Pemotongan

Pola pemotongan yang akan dicari di sini adalah kombinasi strip secara keseluruhan yang mengisi plat. Untuk knapsack I, dipilih di antara strip horisontal-1 atau vertikal-1 dengan profit maksimal. Demikian juga untuk knapsack II, dipilih di antara strip horisontal-2 atau vertikal-2 dengan profit maksimal.



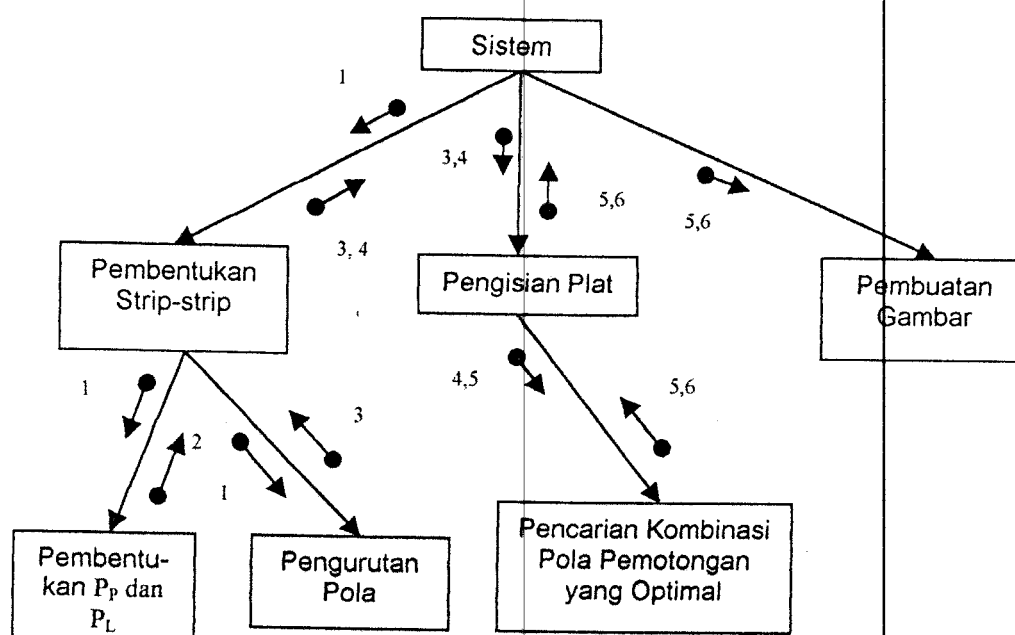
Gambar 4.7 Diagram alir pencarian pola pemotongan

Untuk mencari solusi optimal dilakukan dengan mencari pola pemotongan yang memberi profit maksimal diantara solusi-solusi feasibel tadi. Banyaknya solusi

feasibel adalah $n(P_P \cup P_L)$. Solusi-solusi feasibel ini merupakan kombinasi dari strip-strip vertikal atau horisontal dengan profit yang feasibel untuk plat dan pola-pola tersebut.

4.4.5 Hirarki Modul

Diagram disain modul yang diimplementasikan pada Tugas Akhir ini secara keseluruhan dapat dijabarkan seperti pada gambar 4.8.



Gambar 4.8 Hirarki modul dari sistem

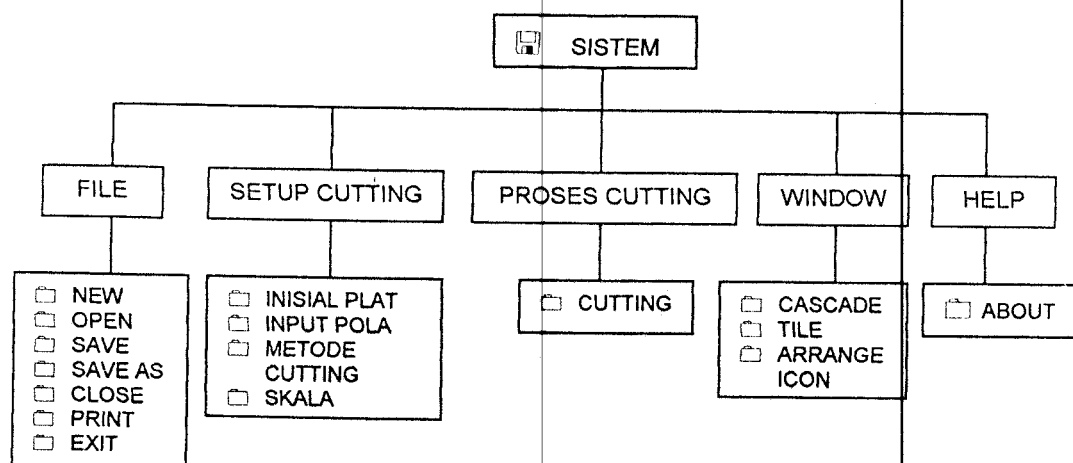
Nomor-nomor di atas mewakili data-data yang masuk dan keluar sesuai dengan diagram data pada gambar 4.2.

1. Data plat dan pola
2. Titik-titik potong guillotine
3. Kumpulan pola terurut

4. Kumpulan strip-strip
5. Solusi-solusi feasibel
6. Solusi optimal

4.5 PERANCANGAN MENU

Untuk dapat memberikan kemudahan dalam pemasukan data dan penggambaran pola potong yang dihasilkan maka pada perangkat lunak ini dirancang antar muka yang menampilkan tiga menu utama yaitu File, Setup Cutting dan Proses Cutting. Menu ini seperti ditunjukkan pada gambar 4.9.



Gambar 4.9 Menu sistem perangkat lunak

Menu File berhubungan dengan berkas teks masukan yaitu berkas pola yang telah pernah dibuat sebelumnya atau untuk menyimpan berkas pola yang baru.

Menu Setup Cutting berhubungan dengan masukan data plat dan data pola yang dimasukkan oleh pengguna. Disini pengguna harus memilih metode pemotongan dengan BSC atau Fase 0-cut. Jika pilihan ini tidak diisi maka proses

dilakukan dengan BSC. Skala dipilih untuk menentukan ukuran skala gambar yang dibuat dari hasil proses yang telah didapatkan.

Menu Proses Cutting berkaitan dengan pilihan metode pemotongan plat yang telah dipilih tadi maka akan dilakukan proses pencarian pola pemotongan optimal yang hasilnya digambarkan pada layar dengan skala pilihan.

Hasil proses yang telah diperoleh maka akan ditampilkan modul informasi yang memberi keterangan tentang hasil tersebut. Informasi terbagi pada dua modul, yang pertama memberikan informasi pada titik-titik normalisasi yang terjadi pemotongan guillotine (horisontal dan vertikal) dan yang kedua terdiri dari Ukuran Bahan yang dipotong, Data Pola-Pola, Sisa yang tidak terpakai dari bahan, Pola-pola Terambil yang optimal, Hasil Proses keseluruhan dan Strip-strip Horisontal dan Vertikal yang dihasilkan sebelum dioptimalkan pada plat awal.

Menu Window dan Help dijelaskan lebih jauh pada lampiran A. Selain menu-menu yang telah disebutkan juga diberikan informasi lainnya. Pada modul pemotongan guillotine, angka ini menunjukkan titik tempat dilakukannya proses guillotine sehingga dihasilkan sub plat dan strip-strip mengisi masing-masing sub plat tersebut. Untuk masing-masing strip merupakan kombinasi dari pola-pola terpilih sehingga mengisi sub plat dengan optimal. Kombinasi dari strip-strip terpilih ini mengisi plat sehingga memberikan solusi feasibel pada titik guillotine tersebut. Untuk keterangan lebih lanjut tentang informasi ini diberikan pada lampiran A.

4.6 PEMBUATAN PERANGKAT LUNAK

Setelah pembahasan perancangan di atas maka berikut ini akan dijelaskan pembuatan dari perangkat lunak itu.

4.6.1 Struktur Data

Untuk menyimpan data plat dan data pola dideklarasikan sebuah tipe data yaitu SzPola yang bertipe rekaman (record). Rekaman ini memiliki enam medan (field) yaitu pola yang merupakan data bertipe larik (array) mewakili ukuran panjang dan lebar pola, panjang dan lebar mewakili panjang dan lebar plat, nPola mewakili banyaknya macam pola dan W menyimpan data panjang atau lebar pola serta V menyimpan nilai serta V masing-masing pola. Untuk W dan V merupakan data larik Data di atas semuanya bertipe integer. Selengkapnya seperti di bawah ini.

```
MaxSubject= Array[0..50] of Integer; {Max & subject}
```

```
SzPola = Record
    Pola : Array[1..50,1..2] of Integer;
    PANJANG,
    LEBAR : Integer;
    nPola:integer;
    W, V :MaxSubject;
end;
```

Untuk menyimpan data strip yang telah didapat dari proses pembentukan strip-strip dibentuk tipe data rekaman baru yang dinamis yaitu InfoStrip. Rekaman ini terdiri lima medan yaitu StrikKe, index, banyak1, banyak2, arah. StripKe ini mewakili urutan strip yang diperoleh, index mewakili pola terurut yang masuk ke dalam strip

```
PtrStrip = ^InfoStrip;
InfoStrip = record
```

```

StripKe, index,
banyak1,
banyak2:integer;
arah : String;
next : ptrStrip; {next pola in infostrip}
end;

```

Data keluaran yang akan ditampilkan sebagai solusi optimal disimpan dalam data rekaman sebagai berikut:

```

PtrPola = ^InfoPola;
InfoPola = record
    StripKe ,index,
    banyak1, banyak2, : integer;
    arah, Guillotine, plat : string;
    next : ptrPola; {next pola in infopola}
end;

```

Medan dari data rekaman ini penjelasannya sama dengan data strip. Guillotine menunjukkan arah pengisian plat dan plat sendiri menunjukkan sub plat yang diisi.

4.6.2 Modul Pembentukan Strip

Berikut yang akan dijabarkan implementasi dari modul pembentukan strip-strip. Alpha dan betha mewakili ukuran dari sub plate yang telah didapat dari pemotongan guillotine.

```

Procedure MembuatStrip(alpha, betha, p, l,  $\pi$ )
  for i  $\leftarrow$  1 to m
    for y  $\leftarrow$  alpha downto 1
      MaxCurrent  $\leftarrow$  0
      Batas  $\leftarrow$  y / W[i]
      if Batas >= 1 then
        for k  $\leftarrow$  1 to Batas
          x  $\leftarrow$  (y-k*W[i])
          if x >= 0 then
            MaxTemp  $\leftarrow$  Tabel[i-1, x] + k*V[i]
            if MaxCurrent < MaxTemp then
              MaxCurrent  $\leftarrow$  MaxTemp
              Frekw  $\leftarrow$  k

```

```

        if MaxCurrent >= Tabel[i,y] then
            Tabel[i,y] ← MaxCurrent
            TabelIndex[i,y] ← i
        Else Tabel[i,y] ← Tabel[i-1, y]
            TabelIndex[i,y] ← 0
        Tabel[i,y] ← Tabel[i-1, y]
        TabelIndex[i,y] ← 0

```

Keterangan:

- MaxCurrent adalah nilai maksimal dari kombinasi pola yang akan membuat pola tersebut terpilih untuk disimpan dalam tabel index, sedangkan MaxCurent disimpan dalam tabel.
- TabelIndex merupakan tabel yang menyimpan data pola terpilih dan frekwensi dari pola tersebut.
- Keluaran dari proses ini adalah tabel index yang berisi pola dan frekwensinya yang membentuk strip-strip.

4.6.3 Modul Pengisian Plat

Pengisian plat dilakukan terhadap plat awal dengan menempatkan strip-strip optimal yang telah didapat. Batas1 dan batas2 mewakili ukuran dari plat.

Algoritma untuk pengisian plat seperti yang telah dijelaskan pada diagram alir gambar 4.6, yang diimplementasikan sebagai berikut:

```

procedure MengisiPlat(var Value: integer, Batas2, Batas1,
n:integer; mode:byte;VH:string);
    inisial_Strip;
    {prosedur program dinamis}
    Dinamis(FMaks,SPola,Batas1);
    {untuk menyimpan hasil proses}
    for i←1 to Spola.nPola do
        y← batas1;
        repeat
            j←TABEL_INDEX[i,y,1]; {berisi pola ke j}

```

```

    if (j<>0) then
        STRIP1[i,j,1] ← j;
        STRIP1[i,j,2] ← TABEL_INDEX[j,y,2];
        y ← (y-STRIP1[i,j,2]*SPola.Pola[j,1]);
        k ← TABEL_INDEX[j,y,2]; {frekw pola berikutnya}
    else
        k ← 0;
    until (k=0)
    for i ← 1 to SPola.nPola do
        SHorizonta1.W[i] ← SPola.Pola[i,2];
        SHorizonta1.V[i] ← FMaks[i];
        Dinamis(Profit,SHorizonta1,Batas2);
        y ← batas2;
        j ← TABEL_INDEX[SPola.nPola,y,1]; {berisi pola ke j}
    repeat
        for q ← 1 to Spola.nPola do
            if (TABEL_INDEX[j,y,2]*STRIP1[j,q,2])>0 then
                ISI_PLATE[j,q,2] ← TABEL_INDEX[j,y,2]*STRIP1[j,q,2];
                ISI_PLATE[j,q,1] ← STRIP1[j,q,1];
                {memasukkan nilai ke dalam Tabel Strip Horizontal}
                if (STRIP1[j,q,2]<>0) then
                    new(nodel);
                    nodel^.StripKe ← j;
                    nodel^.Index ← STRIP1[j,q,1];
                    nodel^.banyak1 ← STRIP1[j,q,2];
                    nodel^.banyak2 ← TABEL_INDEX[j,y,2];
                    nodel^.arah ← 'H';
                    nodel^.next ← nil;
                    if (mode = 1) then
                        INSERT_STRIP(nodel,STRIP_1H[IdxStrip])
                    else
                        INSERT_STRIP(nodel,STRIP_2H[IdxStrip])
                y ← (y-Tabel_Index[j,y,2]*SPola.Pola[j,2]);
                k ← TABEL_INDEX[j,y,2];
                j ← TABEL_INDEX[j,y,1]; {berisi pola ke j}
            until (k=0);
        {menyimpan nilai maksimal dari sub plat}
        Value ← SearchPlate(Profit);

```

Keterangan :

- Hasil dari pembentukan strip-strip (horisontal atau vertikal) dicari sebagai nilai baru untuk tiap pola yang telah terurut di sini.

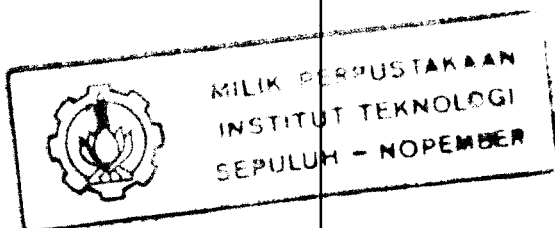
- Kemudian dengan proses dinamis dicari kombinasi strip-strip yang memberi hasil maksimal. Hasil ini berupa pola dan frekwensinya disimpan ke dalam tabel index kembali. Dan nilainya disimpan ke dalam tabel.
- Keluaran dari proses ini adalah tabel index yang berisi kombinasi pola dalam strip-strip dan frekwensi dari strip tersebut yang membentuk solusi feasibel.
- Dari solusi feasibel ini lalu dicari solusi optimal yang memiliki sebaran sisa yang paling sedikit.

4.6.4 Modul Pembuatan Gambar

Untuk menampilkan hasil yang diperoleh berupa solusi optimal maka dilakukan implementasi dengan membaca larik kombinasi pola pemotongan yang didapat dari pengisian plat yaitu sebagai berikut:

```

Procedure BuatGambar;
  MencariNilaiOptimalCuttingPlate(Index);
  PolaT.ListBox1.clear;
  {dari variabel TABLE_POLA[i]}
  if PolaT.Showmodal=mrOk then
    A := '';
    B := '';
    node ← TABLE_POLA[Index];
    repeat
      {plat A}
      if (node^.plate='A') then
        membaca strip-strip yang mengisi sub plat A
      else {plat B}
        membaca strip-strip yang mengisi sub plat A
      node ← node^.next;
    until (node=nil);
    gambarplat;
    gambarPlatA(A);
    gambarPlatB(B);
  
```



BAB V

HASIL UJI COBA DAN PEMBAHASAN

Dalam bab ini akan dibahas tentang uji coba perangkat lunak yang telah diimplentasikan dan evaluasi terhadap hasil dari uji coba tersebut. Uji coba dilakukan terhadap contoh-contoh yang diambil secara *random* dan terhadap contoh-contoh yang telah pernah diuji coba. Hasil yang didapatkan dari uji coba terhadap contoh-contoh tadi dievaluasi di sini untuk melihat kinerja perangkat lunak tersebut.

Permasalahan knapsack dua dimensi yang diselesaikan dengan BSC ini adalah pemotongan terhadap bahan. Uji coba telah dilakukan pada processor 486-DX dan RAM 16 MB dengan menggunakan data random dan data terpilih. Untuk data terpilih merupakan data yang telah diuji coba dalam percobaan pada permasalahan yang sama.

5.1 DATA RANDOM

Untuk uji coba menggunakan data random maka data-data yang digunakan dijelaskan lebih lanjut berikut ini.

Ukuran plat dan banyaknya macam pola m diambil secara random. Ukuran ini dikelompokkan menjadi 2 bagian yaitu ukuran data kecil (ukuran plat antara 50 – 100 dan m antara 10 – 30) dan ukuran data besar (ukuran plat antara 101 – 200 dan m

antara 30 – 50). Data untuk ukuran pola juga dibangkitkan secara random antara ($\frac{1}{4}$ P sampai $\frac{3}{4}$ P) untuk panjangnya dan ($\frac{1}{4}$ L sampai $\frac{3}{4}$ L) untuk lebarnya.

5.1.1 Hasil Komputasi

Uji coba yang telah dilakukan terhadap data random di sini menggunakan data plat (100, 100) untuk ukuran kecil dengan banyaknya macam pola (m) 10, 20, 30, 40 dan 50. Untuk tiap m dilakukan 5 kali uji coba dengan plat yang sama. Sedangkan untuk data ukuran besar plat yang digunakan berukuran (200, 200) dan macam pola sama dengan yang sebelumnya. Perulangan terhadap tiap pola yang dipakai juga sebanyak 5 kali. Data random yang diuji coba dapat dilihat pada lampiran B.

Hasil uji coba terhadap data random ini didapatkan hasil seperti yang dicantumkan dalam tabel 5.1 dan tabel 5.2 .

Tabel 5.1 Hasil BSC dan Fase 0_Cut
(ukuran data kecil)

Ukuran Plat (P, L)	Jumlah Pola (m)	Solusi Optimal		BSC
		BSC	Fase 0-Cut	Rata-Rata
(100,100)	10	93%	93%	89%
	20	97%	96%	93%
	30	99%	98%	95%
	40	99%	98%	96%
	50	100%	99%	97%

Tabel 5.2 Hasil BSC dan Fase 0_Cut
(ukuran data besar)

Ukuran Plat (P, L)	Jumlah Pola (m)	Solusi Optimal		BSC
		BSC	Rata-Rata	Rata-Rata
(200, 200)	10	94%	93%	87%
	20	96%	94%	91%
	30	98%	97%	94%
	40	98%	97%	95%
	50	98%	98%	95%

Evaluasi dilakukan terhadap output yang dihasilkan dari uji coba terhadap contoh-contoh di atas. Untuk contoh-contoh dari data random, pada tabel 5.1 di atas memperlihatkan hasil dari rasio aproksimasi yang baik. Untuk data ukuran kecil, rasio aproksimasi ini semakin meningkat dengan bertambah banyaknya pola. Persentase rata-rata solusi optimal untuk ukuran data kecil adalah 96% dengan waktu yang dibutuhkan untuk mendapatkan solusi optimal 3552 mdetik sedangkan untuk fase 0-cut persentase rata-rata solusi optimal adalah 95,7% dengan waktu 1595 mdetik.

Sedangkan untuk data ukuran besar rasio aproksimasi kecenderungannya juga semakin meningkat untuk bertambahnya macam pola. Untuk hasil-hasil yang ditunjukkan pada tabel 5.2. ini menunjukkan hasil yang cukup baik sebagai solusi optimal. Persentase rata-rata solusi optimal untuk ukuran data besar adalah 98% dengan waktu yang dibutuhkan untuk mendapatkan solusi optimal 5070 mdetik sedangkan untuk fase 0-cut persentase rata-rata solusi optimal adalah 97% dengan waktu 1884 mdetik.

Jika hasil yang ditunjukkan dari fase 0-cut dipertimbangkan maka rasio aproksimasi (persentase solusi optimal) juga memperlihatkan solusi cukup baik yang optimal. Jadi penyelesaian dengan BSC juga dapat dihentikan jika dipertimbangkan bahwa solusi optimal telah didapatkan dari solusi yang diberikan oleh fase 0-cut ini. Pertimbangan lain juga dengan fase 0-cut ini hanya membutuhkan waktu yang relatif lebih kecil dari segi komputasinya.

5.2 DATA TERPILIH

Data terpilih adalah data yang diambil dari percobaan yang telah pernah dilakukan orang terhadap permasalahan yang sama. Data yang disajikan ada 4 macam data dalam tabel, yang dianggap dapat mewakili dari permasalahan.

Tabel 5.3 Contoh 1¹

Plat (P, L)	Pola (p, l)
(9, 11)	(3, 3), (4, 3), (2, 4)

Tabel 5.4 Contoh 2²

Plat (P, L)	Pola (p, l)
(15, 10)	(8, 4), (3, 7), (8, 2), (3, 4), (3, 3), (3, 2), (2, 1)

¹ Fayard D., Zissimopoulos V., *An Algorithm for Solving Unconstrained Two-Dimensional Knapsack Problems*, European Journal of Operational Research (1995), 621.

² Christofides N., Whitock C., *An Algorithm for Two-Dimensional Cutting Problems*, Operations Research, Vol 25, No. 1, Jan-Feb 1977, 41.

Tabel 5.5 Contoh 3³

Plat (P, L)	Pola (p, l)
(40, 70)	(9, 24), (10, 14), (12, 8), (9, 35), (21, 22), (11, 13), (31, 13), (30, 7), (14, 8), (13, 7)

Tabel 5.6 Contoh 4⁴

Plat (P, L)	Pola (p, l)
(127, 98)	(21, 13), (36, 17), (54, 20), (20, 27), (18, 65)

Tabel 5.7 Contoh 5⁵

Plat (P, L)	Pola (p, l)
(40, 70)	(31,43), (30,41), (29,39), (28,38), (27,37), (26,36), (25,35), (24,34), (33,23), (22,32), (31,21), (29,18), (17,27), (15,24), (16,25), (15,24), (23,14), (21,12), (19,11), (9,17)

Untuk data terpilih ini yang akan dibahas lebih lanjut adalah data dari contoh_1. Untuk contoh yang lainnya hanya disajikan sebagai hasil perhitungan akhir saja.

Diberikan sebuah lembaran plat dengan ukuran panjang dan lebarnya adalah 9 x 11 yang digunakan untuk memenuhi order dengan pola-pola 3 x 3, 4 x 3 dan 2 x 4. Hitunglah profit pola potong untuk mendapatkan solusi yang optimal ?

Data-data:

Ukuran Plat : Panjang = 9, Lebar = 11

Data Pola : Himpunan pola (p_i, l_i) : $S = \{(3,3), (4,3), (2,4)\}$

Nilai : $\pi_i = p_i \times l_i$

³ Ibid, 41

⁴ Fayard D., op. cit., hal 630

⁵ Chistofide N., op. cit., hal 41

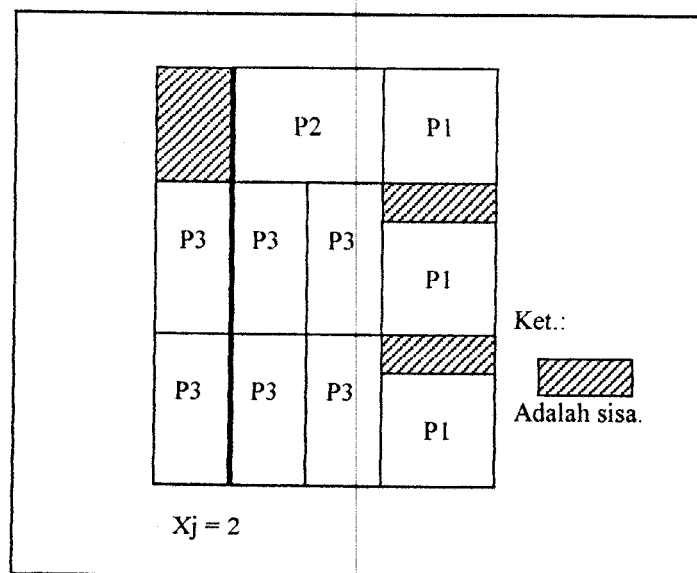
5.2.1 Penyelesaian Dengan BSC

Titik-titik normalisasi yang diperoleh $x_j \in P_p = \{2, 3, 4\}$ dan $x_j \in P_l = \{3, 4\}$. Pada masing-masing titik ini diperoleh solusi-solusi feasibel dengan kombinasi pola. Solusi feasibel yang diperoleh pada saat dilakukan guillotine vertikal disajikan dalam tabel di bawah ini sebagai berikut:

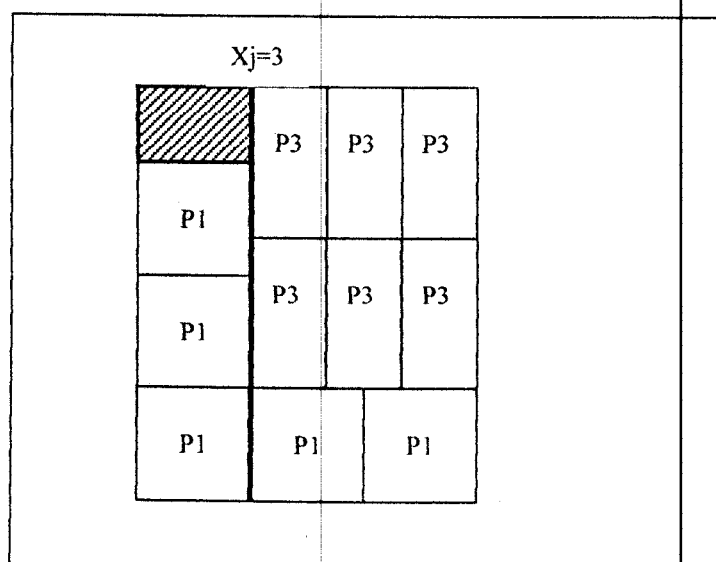
Tabel 5.8 Solusi Feasibel pada Guillotine Vertikal

No.	Titik potong guillotine vertikal	Solusi feasibel	Sisa
1	2	87	12
2	3	93	6
3	4	90	9

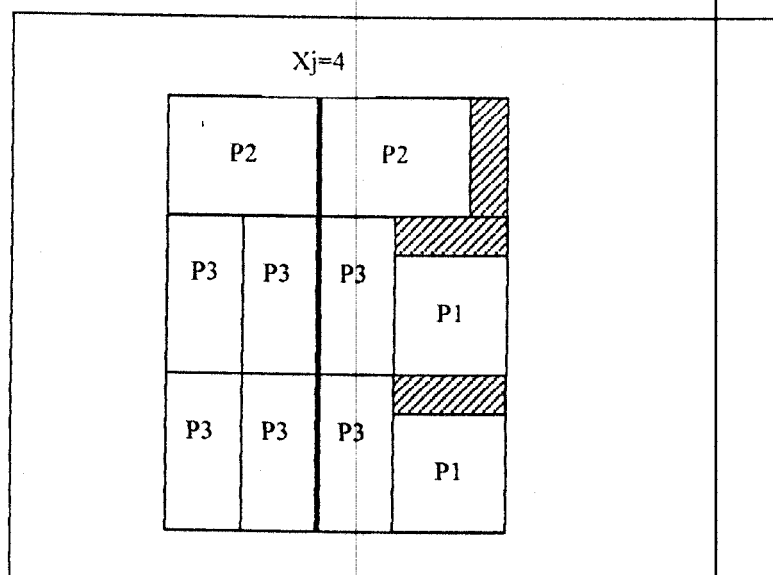
Dari solusi feasibel ini digambarkan bagaimana susunan dari kombinasi pola pemotongan tersebut untuk masing-masing titik potong tersebut.



Gambar 5.1 Solusi feasibel 1 untuk plat (9,11)



Gambar 5.2 Solusi feasibel 2 untuk plat (9,11)



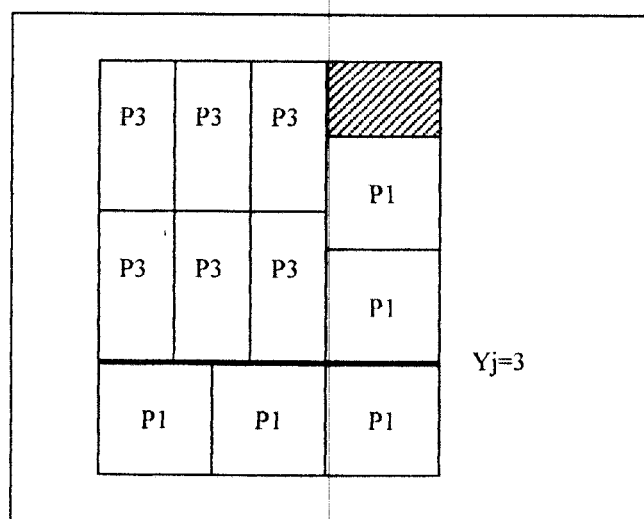
Gambar 5.3 Solusi feasibel 3 untuk plat (9,11)

Solusi feasibel pada saat dilakukan guillotine horisontal adalah sebagai berikut:

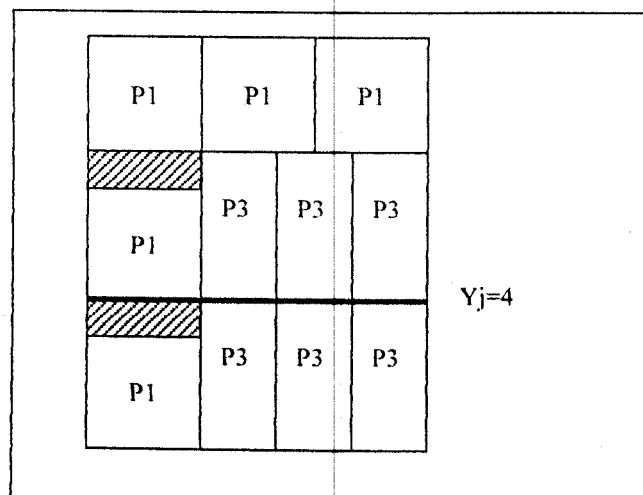
Tabel 5.9 Solusi feasibel pada guillotine horisontal

No.	Titik potong guillotine horisontal	Solusi feasibel	Sisa
1	3	93	6
2	4	93	6

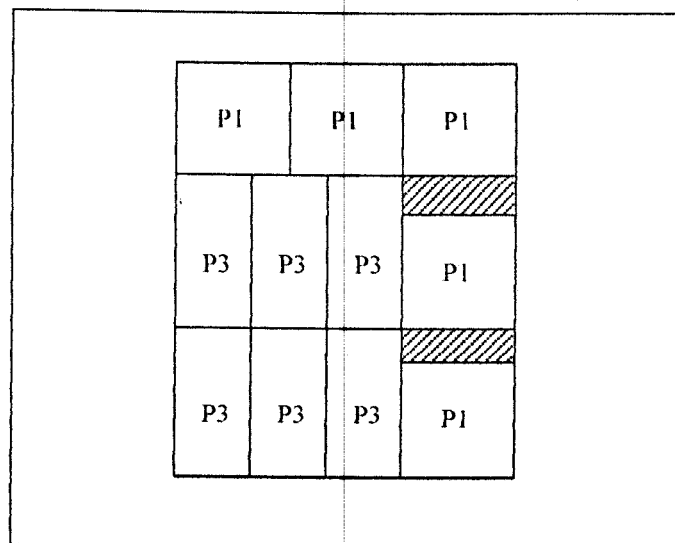
Dari solusi feasibel ini digambarkan bagaimana susunan dari kombinasi pola potong tersebut untuk masing-masing titik potong tersebut.



Gambar 5.4 Solusi feasibel 4 untuk plat (9,11)



Gambar 5.5 Solusi feasibel 5 untuk plat (9,11)



Gambar 5.6 Solusi feasibel fase 0-cut

5.2.2 Fase 0_Cut

Pada fase 0-cut, penyelesaian masalah dilakukan tanpa pemotongan guillotine di titik normalisasi pada plat awal seperti pada bagian 5.2.1. Jadi di sini pencarian dilakukan pada kapasitas P dan L. Hasil yang diperoleh yaitu nilai

profitnya adalah 93 dan sisa 6. Kombinasi pola disusun oleh 2 buah strip horisontal yang merupakan kombinasi pola-1 (3 x 3) dan pola-3 (2 x 4) dan sebuah strip horisontal yang tersusun dari 3 buah pola-1 (3 x 3).

Kombinasi pola pemotongan pada fase 0-cut dapat dilihat pada gambar 5.6 di atas.

5.2.3 Hasil Komputasi

Untuk hasil uji coba terhadap data terpilih contoh 1 sampai contoh 4 dengan menggunakan BSC dan fase 0-cut didapatkan:

Tabel 5.10 Hasil BSC dan Fase 0_Cut

Plat (P, L)	m	Solusi Sebe- lumnya	Solusi Optimal				BSC Rata- Rata
			BSC		Fase 0-cut		
			Nilai	Waktu (dtk)	Nilai	Waktu (dtk)	
(9,11)	3	93	93 (94%)	0,110	93 (94%)	0,060	92%
(15, 10)	7	249	150 (100%)	0,170	150 (100%)	0,060	100%
(40, 70)	10	3006	2800 (100%)	0,930	2800 (100%)	0,060	99%
(127, 98)	5	12192	12192 (98%)	0,870	12132 (97%)	0,050	95%
(40, 70)	20	2240	2694 (96%)	1,100	2694 (96%)	0,110	93%

Dari evaluasi terhadap output data terpilih pada tabel 5.10, ditunjukkan bahwa untuk contoh 2, dengan BSC memberi solusi 150(100%) dengan waktu 0,17 detik. Sedangkan solusi yang diberikan dengan algoritma *tree-search* adalah 249 dalam waktu 2,47 detik.

Untuk contoh 3, BSC memberikan solusi 2800(100%) dalam waktu 0,93 detik sedangkan dengan algoritma *tree search* solusinya 3006 dalam waktu 24,07

detik. Untuk dua contoh ini BSC memberikan solusi optimal tanpa sisa bahan dalam waktu sepersekian detik. Algoritma *tree search* membutuhkan waktu relatif lebih lama untuk menemukan solusi optimalnya. Dalam kedua contoh ini algoritma mendefinisikan nilai pola bukan berdasarkan pada luasannya. Dengan menggunakan fase 0-cut solusi yang diberikan sama dengan hasil dengan BSC dalam waktu yang relatif lebih singkat dari BSC.

Untuk contoh 4, BSC memberikan solusi sebesar 12192(98%) dalam waktu 0,87 detik. Fase 0-cut memberikan solusi sebesar 12132(97%). Pada contoh ini hasil BSC lebih baik dari fase 0-cut. Untuk contoh 5, hasil yang diberikan oleh algoritma *exact* 2240 dalam 4,51 detik dan dengan BSC diperoleh solusi optimal 2694(96%) dalam waktu 1,1 detik. Hasil yang diberikan oleh BSC lebih baik dari algoritma *exact* untuk contoh ini.

BAB VI

PENUTUP

6.1 KESIMPULAN

Dari evaluasi terhadap hasil uji coba memberi suatu gambaran yang dapat disimpulkan sebagai berikut:

1. Dari hasil uji coba untuk ukuran data kecil (banyaknya macam pola antara 10 - 30 dan ukuran plat antara 50 - 100) dan ukuran data besar (banyaknya macam pola antara 30 – 50 dan ukuran plat antara 101 - 200) menunjukkan BSC adalah efisien jika digunakan pada data ukuran besar.
2. Rasio aproksimasi terbaik yang diberikan oleh BSC untuk ukuran data kecil adalah 96% dan untuk ukuran data besar adalah 98%.
3. Fase 0-cut memberikan solusi yang cukup baik juga untuk permasalahan knapsack dua dimensi yang sama, sehingga dapat dipertimbangkan untuk penyelesaian masalah knapsack ini.
4. Dengan menggunakan program dinamis jumlah knapsack yang dipakai hanya 4 buah knapsack saja untuk tiap titik normalisasi, sehingga akan mengurangi memori yang terpakai.

6.2 SARAN

Algoritma aproksimasi yang diimplementasikan guna penyelesaian masalah knapsack dua dimensi ini dapat dikembangkan lebih lanjut, mengingat algoritma ini dapat menangani dengan efisien data dengan ukuran besar.

1. Algoritma aproksimasi ini selanjutnya dapat dikembangkan lagi dengan menggunakan nilai untuk setiap pola bukan hanya dihubungkan dengan luasan pola itu, misalnya dengan prioritas atau profit dari pola itu yang dapat didefinisikan sendiri.
2. Algoritma ini dapat dikembangkan untuk diterapkan pada penyelesaian masalah knapsack yang terbatas (bounded). Di masalah ini tentunya perlu memperhatikan batas maksimum yang harus dipenuhi dari tiap pola dan jumlah plat yang tersedia. Jumlah masing-masing item pola yang dibutuhkan dibatasi jumlahnya. Sehingga model matematikanya disesuaikan dengan batasan yang baru. Dalam tugas akhir ini variabel k ($= \lfloor P/p \rfloor$ atau $= \lfloor L/l \rfloor$) dapat dijadikan acuan untuk mendapatkan item pola sejumlah yang dibutuhkan. Pengembangan ini juga dapat digabungkan dengan saran sebelumnya.

DAFTAR PUSTAKA

- 1) Fayard D., Zissimopoulos V., **"An Approximation Algorithm For Solving Unconstrained Two-Dimensional Knapsack Problems"**, European Journal of Operational Research (1995) 618-632.
- 2) J. E. Beasley, **"Algorithms for Unconstrained Two-dimensional Guillotine Cutting"**, Operational Research Society (1985), Vol 36, No. 4, 297-306.
- 3) Martello S., Toth P., **"Knapsack Problems: Algorithms and Computer Implementations"**, John Willey & Sons, 1990.
- 4) Pratono A., **"Perancangan dan Pembuatan Perangkat Lunak Optimasi Benda 2 Dimensi"**, Teknik Komputer, ITS, 1995.
- 5) R. G. Dyson, A. S. Gregory, **"The Cutting Stock Problem in Flat Glass Industry"**, Operational Research Quartely, Vol 25 No. 1, 41-53.
- 6) Smith D. K., **"Dynamic Programming, A Practical Introduction"**, Ellis Horwood Ltd., 1991.
- 7) Taha, H. A., **"Riset Operasional"**, Binarupa Aksara, 1996.

LAMPIRAN A

PANDUAN PENGOPERASIAN PERANGKAT LUNAK

Untuk dapat menjalankan aplikasi perangkat lunak ini, di sini akan dijelaskan kebutuhan perangkat lunak dan perangkat keras yang harus diperhatikan sebelumnya.

A.1. Kebutuhan Perangkat Lunak dan Keras

Perangkat Lunak ini harus dijalankan dalam sistem operasi windows dengan minimal kebutuhan perangkat keras yang harus disediakan adalah:

- CPU 486-DX
- Monitor VGA
- Memori 16 MB
- HD 80 MB

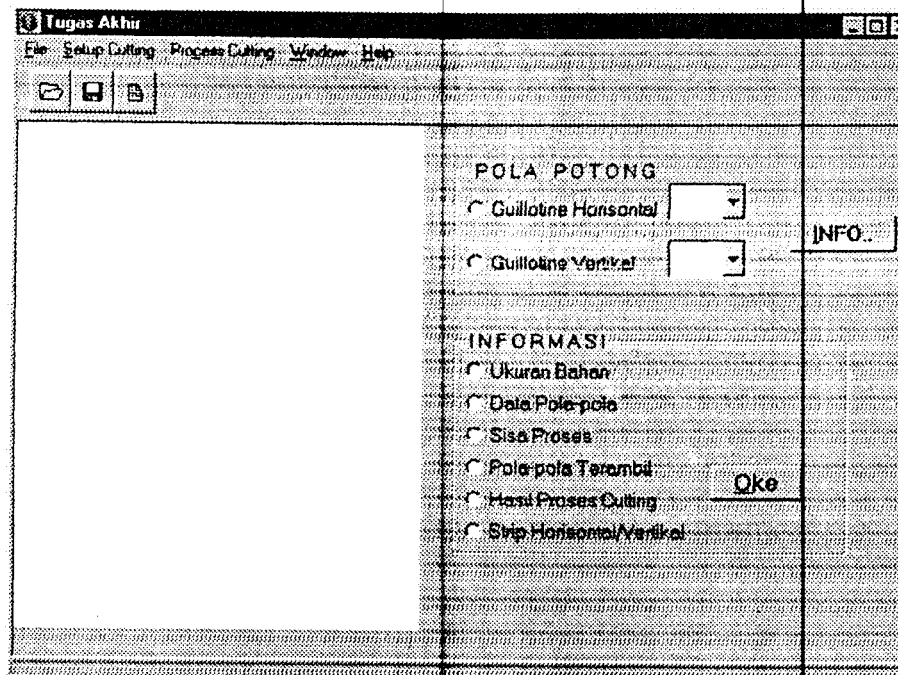
Sedangkan untuk kebutuhan perangkat lunak adalah sebagai berikut:

- Sistem Operasi Windows 95
- File-file sebagai berikut : FINAL.EXE, MDIMAIN.PAS,
IN_POLA.PAS, POLAS.PAS.

A.2. Cara Pengoperasian Perangkat Lunak

Untuk menjalankan perangkat lunak ini pilih FINAL untuk dieksekusi. Tampilan yang muncul adalah menu utama dari perangkat lunak ini. Seperti terlihat pada gambar A.1.

Menu yang tersedia di sini yaitu, File, Setup Cutting, Proses Cutting, Window dan Help.



Gambar A.1 Tampilan menu utama

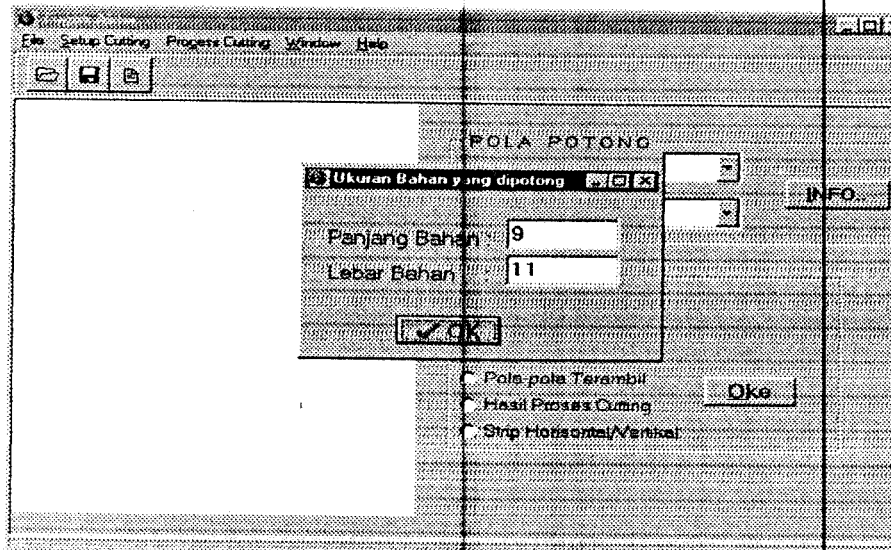
Item-item dan Fungsi dari menu utama dijelaskan dalam tabel berikut ini.

Kelompok	Menu Item	Fungsi
File	New	Membersihkan layar
	Open	Membuka file data
	Save	Menyimpan file
	Save As	Menyimpan file dengan nama lain
	Close	Menutup Window child
	Print	Mencetak ke printer
	Exit	Keluar dari aplikasi
Setup Cutting	Inisial Dipotong	Memasukkan data plat
	Input Pola	Memasukkan data pola
	Metode Cutting	Memilih cara pemotongan
Proses Cutting	Cutting	Melakukan proses pemotongan
Window	Cascade	Mengatur susunan window bertumpuk
	Tile	Mengatur susunan window paralel
	Arrange Icon	Mengatur susunan window icon minimum
Help	About	Menampilkan nama dan topik

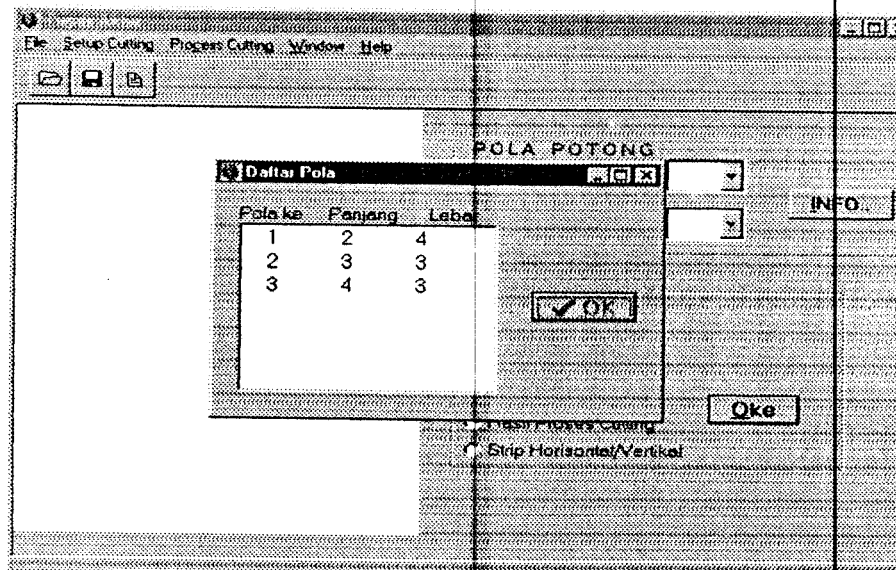
Tabel A.1. Fasilitas fungsi menu pengoperasian perangkat lunak

A.3. Menyiapkan Data Untuk Proses

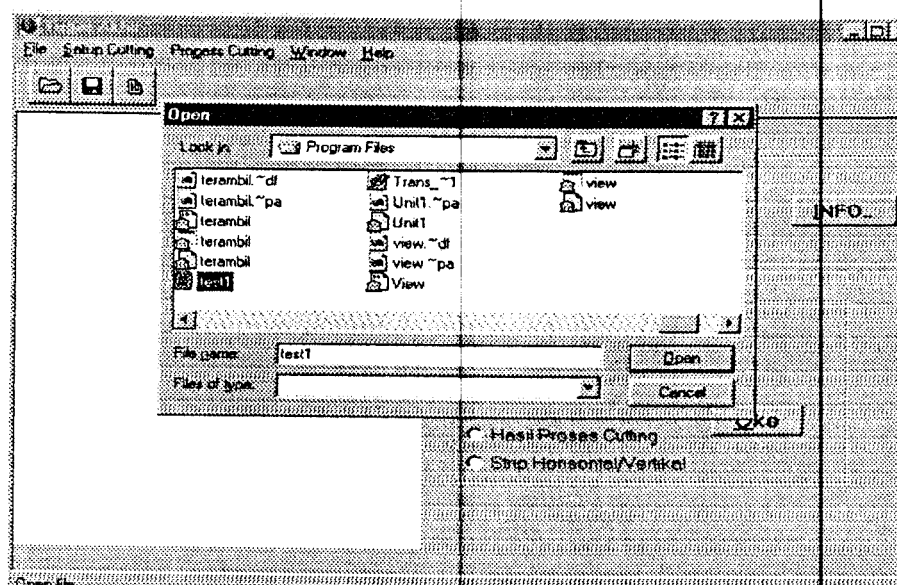
Data-data input yang diperlukan terdiri atas dua macam yaitu data plat dan data pola. Untuk data plat diinputkan melalui menu Setup Cutting kemudian memilih Inisial Cutting. Sedangkan data plat dapat diinputkan melalui menu yang sama lalu memilih Input Pola. Selain dengan memasukkan langsung data pola juga diambil dari file data yang telah disiapkan.



Gambar A.2 Memasukkan data plat



Gambar A.3 Memasukkan data pola



Gambar A.4 Membuka file data pola

A.4. Melakukan Proses Pemotongan

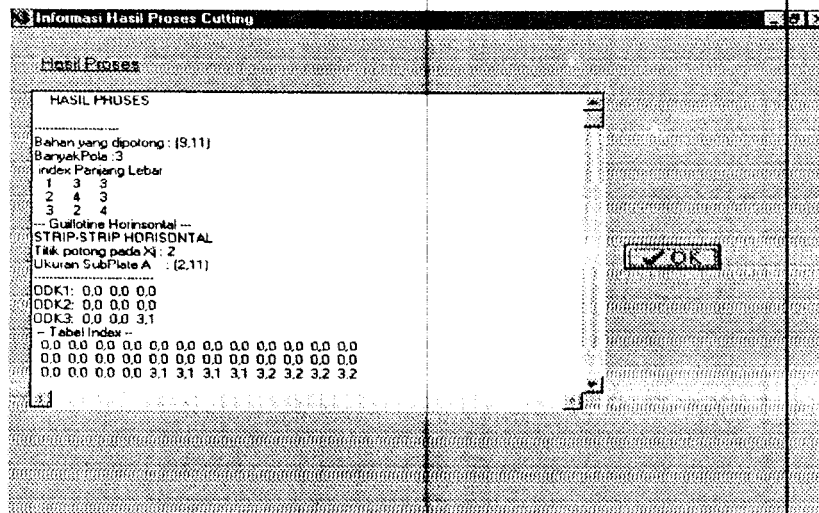
Menu yang harus dipilih untuk proses pemotongan adalah Proses Cutting dengan memilih Cutting. Proses akan berjalan untuk menghasilkan kombinasi pola pemotongan optimal. Hasil ini disimpan dalam file *<hasil.txt>*, yang dapat dilihat kembali melalui modul Informasi yang disediakan.

A.5. Melihat Informasi Hasil Proses Pemotongan

Informasi-informasi yang dapat diperlihatkan adalah sebagai berikut:

A.5.1. Informasi Dari Setiap Proses

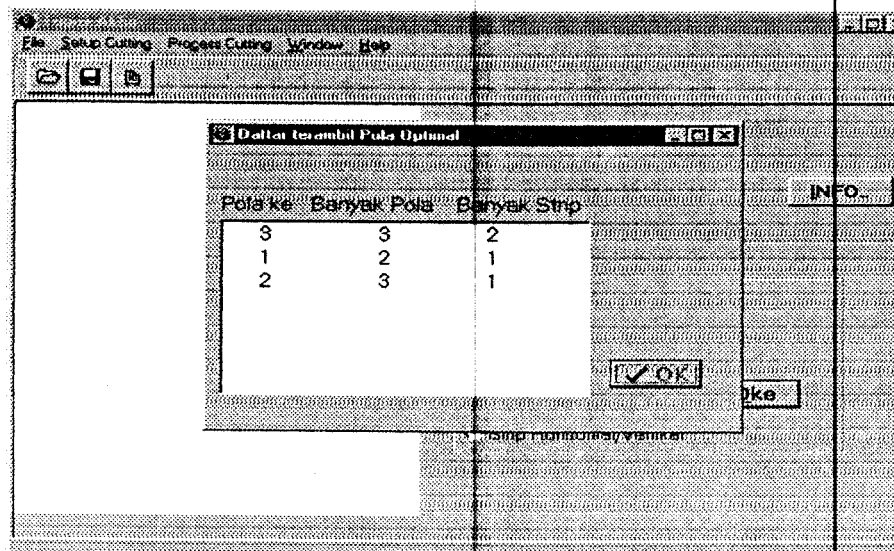
Informasi yang disediakan selain data plat, data pola juga semua kombinasi yang dihasilkan dari proses pemotongan pada tiap titik normalisasi ditampilkan di sini.



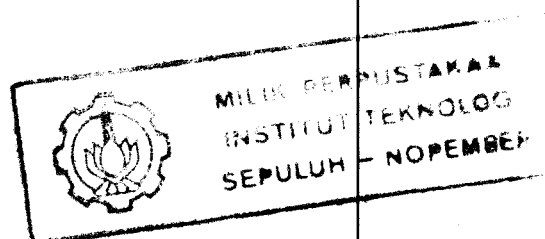
Gambar A.5 Menampilkan file hasil proses

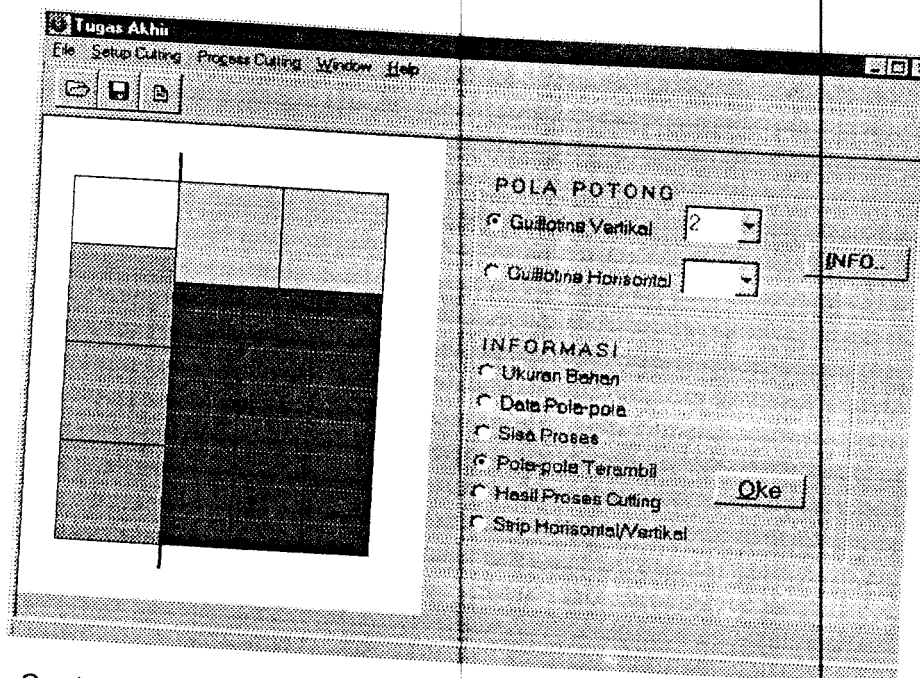
A.5.2. Kombinasi Pola Pemotongan Optimal

Informasi yang diberikan adalah kombinasi pola pemotongan optimal dengan memberikan keterangan jumlah pola terambil masing-masing dan gambar dari kombinasi pola pada plat.



Gambar A.6 Menampilkan kombinasi pola pemotongan optimal

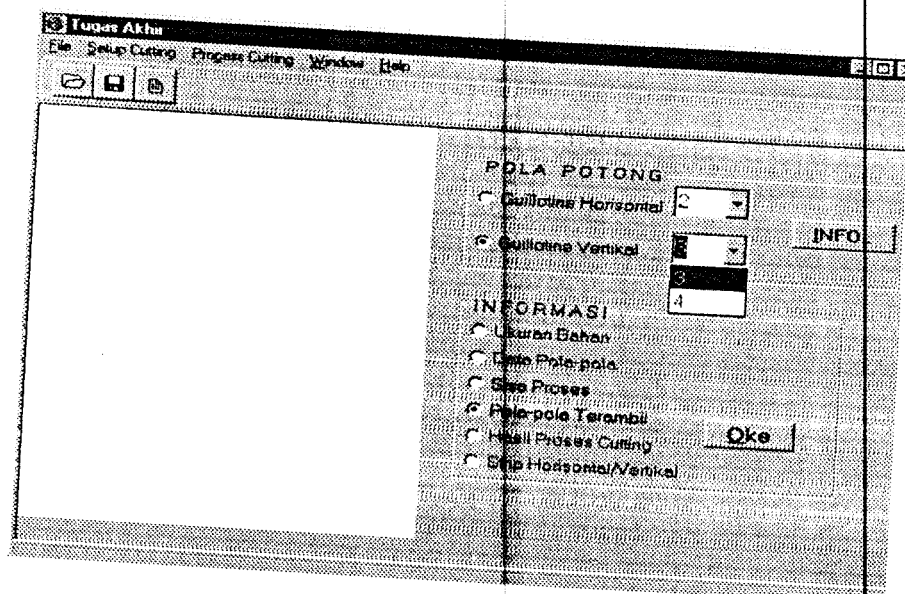




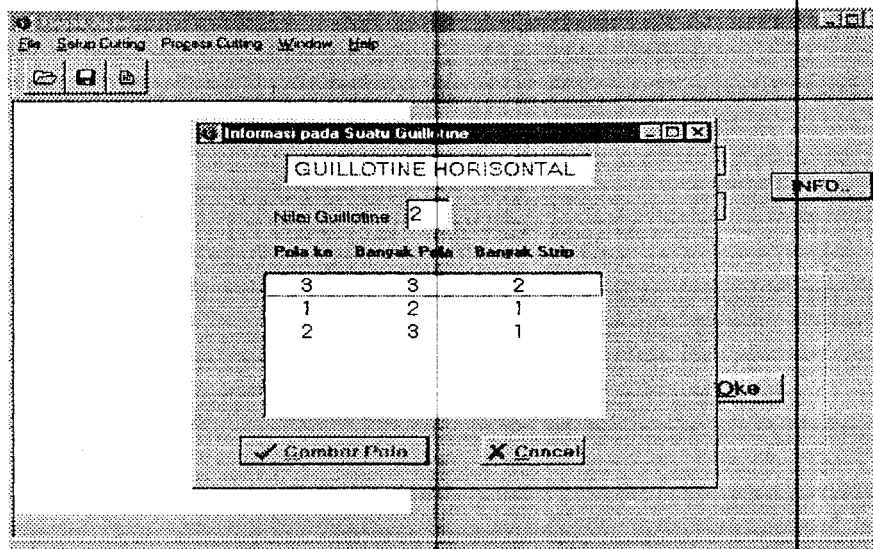
Gambar A.7 Menampilkan gambar kombinasi pola pemotongan optimal

A.5.3. *Pemilihan Guillotine Horizontal Pada Suatu Titik Normalisasi*

Informasi yang akan diberikan jika masuk ke modul ini adalah dengan memilih titik potong normalisasi arah horisontal maka akan disajikan kombinasi pola pemotongan dan gambar pola pemotongan juga.



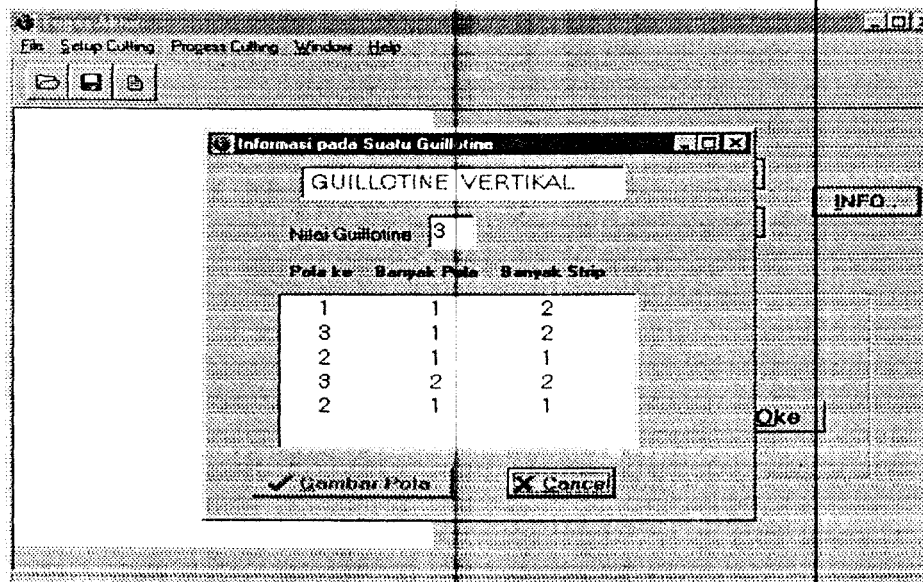
Gambar A.8a. Menampilkan pilihan titik guillotine horisontal



Gambar A.8b. Menampilkan guillotine horisontal

A.5.4. Pemilihan Guillotine Vertikal Pada Suatu Titik Normalisasi

Informasi yang diberikan dari modul ini sama dengan di atas hanya untuk arah pemotongan guillotine vertikal.



Gambar A.9 Menampilkan guillotine vertikal

LAMPIRAN B

DATA INPUT RANDOM

B.1 Data Ukuran Kecil

Ukuran Plat (P, L) = (100, 100); Ukuran Pola (p, l)

Macam Pola = m (antara 10 sampai 50)

Plat (P, L)	m	Pola (p, l)
(100, 100)	10	(25, 26) (68, 35) (38, 58) (40, 33) (43, 46) (29, 48) (28, 67) (27, 39) (70, 43) (63, 41)
	10	(59, 67) (60, 40) (33, 41) (48, 37) (66, 38) (49, 32) (68, 39) (63, 73) (49, 69) (66, 26)
	10	(32, 32) (50, 26) (54, 25) (63, 57) (63, 60) (52, 35) (59, 54) (72, 57) (74, 37) (58, 39)
	10	(29, 63) (49, 69) (50, 53) (72, 59) (41, 25) (60, 74) (63, 62) (34, 32) (70, 64) (54, 68)
	10	(59, 39) (29, 56) (39, 68) (33, 38) (52, 73) (33, 32) (38, 33) (53, 49) (39, 32) (55, 61)

Plat (P, L)	m	Pola (p, l)
(100, 100)	20	(25, 26) (68, 35) (38, 58) (40, 33) (43, 46) (29, 48) (28, 67) (27, 9) (70, 43) (63, 41) (59, 67) (60, 40) (33, 41) (48, 37) (66, 38) (49, 32) (68, 39) (63, 73) (49, 69) (66, 26)
	20	(32, 32) (50, 26) (54, 25) (63, 57) (63, 60) (52, 35) (59, 54) (72, 7) (74, 37) (58, 39) (29, 63) (49, 69) (50, 53) (72, 59) (41, 25) (60, 4) (63, 62) (34, 32) (70, 64) (54, 68)
	20	(59, 39) (29, 56) (39, 68) (33, 38) (52, 73) (33, 32) (38, 33) (53, 9) (39, 32) (55, 61) (62, 29) (28, 27) (30, 64) (65, 44) (55, 46) (62, 9) (42, 65) (65, 32) (29, 28) (28, 34)
	20	(58, 36) (53, 37) (43, 66) (40, 40) (57, 59) (70, 40) (48, 39) (63, 62) (52, 61) (45, 52) (54, 69) (58, 30) (48, 45) (30, 27) (52, 58) (29, 71) (66, 59) (71, 38) (60, 61) (49, 58)
	20	(32, 36) (72, 43) (31, 74) (40, 27) (30, 68) (52, 59) (30, 58) (25, 40) (70, 51) (41, 47) (70, 42) (42, 49) (28, 67) (63, 34) (71, 49) (31, 9) (45, 35) (65, 54) (30, 56) (71, 52)

Plat (P, L)	m	Pola (p, l)
(100, 100)	30	(25, 26) (68, 35) (38, 58) (40, 33) (43, 46) (29, 48) (28, 67) (27, 39) (70, 43) (63, 41) (59, 67) (60, 40) (33, 41) (48, 37) (66, 38) (49, 32) (68, 39) (63, 73) (49, 69) (66, 26) (32, 32) (50, 26) (54, 25) (63, 7) (63, 60) (52, 35) (59, 54) (72, 57) (74, 37) (58, 39)

	30	(29, 63) (49, 69) (50, 53) (72, 59) (41, 25) (60, 74) (63, 62) (34, 32) (70, 64) (54, 68) (59, 39) (29, 56) (39, 68) (33, 38) (52, 73) (33, 32) (38, 33) (53, 49) (39, 32) (55, 61) (62, 29) (28, 27) (30, 64) (65, 44) (55, 46) (62, 49) (42, 65) (65, 32) (29, 28) (28, 34)	
	30	(58, 36) (53, 37) (43, 66) (40, 40) (57, 59) (70, 40) (48, 39) (63, 2) (52, 61) (45, 52) (54, 69) (58, 30) (48, 45) (30, 27) (52, 58) (29, 71) (66, 59) (71, 38) (60, 61) (49, 58) (32, 36) (72, 43) (31, 74) (40, 27) (30, 68) (52, 59) (30, 58) (25, 40) (70, 51) (41, 47)	
	30	(70, 42) (42, 49) (28, 67) (63, 34) (71, 49) (31, 69) (45, 35) (65, 54) (30, 56) (71, 52) (41, 47) (36, 34) (59, 74) (56, 59) (52, 29) (62, 32) (36, 69) (60, 46) (74, 32) (26, 34) (25, 43) (44, 43) (29, 47) (50, 48) (58, 65) (53, 37) (43, 54) (68, 55) (37, 59) (52, 68)	
	30	(43, 43) (56, 64) (47, 35) (59, 66) (62, 70) (34, 63) (38, 39) (60, 47) (73, 34) (25, 36) (69, 41) (68, 63) (73, 38) (44, 66) (26, 37) (27, 50) (46, 25) (32, 60) (40, 44) (52, 38) (32, 39) (28, 60) (41, 64) (57, 53) (69, 26) (37, 69) (59, 26) (32, 32) (45, 45) (32, 25)	

Plat (P, L)	M	Pola (p, l)	
(100, 100)	40	(25, 26) (68, 35) (38, 58) (40, 33) (43, 46) (63, 41) (59, 67) (60, 40) (33, 41) (48, 37) (66, 38) (49, 32) (68, 39) (63, 73) (49, 69) (66, 26) (32, 32) (50, 26) (54, 25) (63, 57) (63, 60) (52, 35) (59, 54) (72, 57) (74, 37) (58, 39) (29, 63) (49, 69) (50, 53) (72, 59) (41, 25) (60, 74) (63, 62) (34, 32) (70, 64) (54, 68)	
	40	(59, 39) (29, 56) (39, 68) (33, 38) (52, 73) (33, 32) (38, 33) (53, 49) (39, 32) (55, 61) (62, 29) (28, 27) (30, 64) (65, 44) (55, 46) (62, 49) (42, 65) (65, 32) (29, 28) (28, 34) (58, 36) (53, 37) (43, 66) (40, 40) (57, 59) (70, 40) (48, 39) (63, 62) (52, 61) (45, 52) (54, 69) (58, 30) (48, 45) (30, 27) (52, 58) (29, 71) (66, 59) (71, 38) (60, 61) (49, 58)	
	40	(32, 36) (72, 43) (31, 74) (40, 27) (30, 68) (52, 59) (30, 58) (25, 40) (70, 51) (41, 47) (70, 42) (42, 49) (28, 67) (63, 34) (71, 49) (31, 69) (45, 35) (65, 54) (30, 56) (71, 52) (41, 47) (36, 34) (59, 74) (56, 59) (52, 29) (62, 32) (36, 69) (60, 46) (74, 32) (26, 34) (25, 43) (44, 43) (29, 47) (50, 48) (58, 65) (53, 37) (43, 54) (68, 55) (37, 59) (52, 68)	
	40	(43, 43) (56, 64) (47, 35) (59, 66) (62, 70) (34, 63) (38, 39) (60, 47) (73, 34) (25, 36) (69, 41) (68, 63) (73, 38) (44, 66) (26, 37) (27, 50) (46, 25) (32, 60) (40, 44) (52, 38) (32, 39) (28, 60) (41, 64) (57, 53) (69, 26) (37, 69) (59, 26) (32, 32) (45, 45) (32, 25) (51, 63) (53, 36) (68, 28) (49, 28) (43, 58) (35, 65) (51, 45) (26, 74) (33, 27) (27, 30)	
	40	(65, 72) (30, 35) (63, 74) (49, 26) (54, 65) (71, 50) (58, 74) (63, 60) (69, 74) (61, 64) (25, 43) (40, 28) (64, 38) (67, 70) (27, 69) (71, 9) (59, 43) (46, 62) (59, 42) (56, 50) (73, 30) (74, 26) (55, 42) (50, 57) (65, 56) (72, 49) (49, 74) (36, 34) (46, 59) (62, 26) (55, 65) (48, 33) (56, 28) (73, 64) (68, 46) (25, 36) (53, 69) (27, 48) (71, 68) (67, 61)	

Plat (P, L)	M	Pola (p, l)
(100, 100)	50	(25, 26) (68, 35) (38, 58) (40, 33) (43, 46) (29, 48) (28, 67) (27, 39) (70, 43) (63, 41) (59, 67) (60, 40) (33, 41) (48, 37) (66, 38) (49, 32) (68, 39) (63, 73) (49, 69) (66, 26) (32, 32) (50, 26) (54, 25) (63, 57) (63, 60) (52, 35) (59, 54) (72, 57) (74, 37) (58, 39) (29, 63) (49, 69) (50, 53) (72, 59) (41, 25) (60, 74) (63, 62) (34, 32) (70, 64) (54, 68) (59, 39) (29, 56) (39, 68) (33, 38) (52, 73) (33, 32) (38, 33) (53, 49) (39, 32) (55, 61)
	50	(62, 29) (28, 27) (30, 64) (65, 44) (55, 46) (62, 49) (42, 65) (65, 32) (29, 28) (28, 34) (58, 36) (53, 37) (43, 66) (40, 40) (57, 59) (70, 40) (48, 39) (63, 62) (52, 61) (45, 52) (54, 69) (58, 30) (48, 45) (30, 27) (52, 58) (29, 71) (66, 59) (71, 38) (60, 61) (49, 58) (32, 36) (72, 43) (31, 74) (40, 27) (30, 68) (52, 59) (30, 58) (25, 40) (70, 51) (41, 47) (70, 42) (42, 49) (28, 67) (63, 34) (71, 49) (31, 69) (45, 35) (65, 54) (30, 56) (71, 52)
	50	(41, 47) (36, 34) (59, 74) (56, 59) (52, 29) (62, 32) (36, 69) (60, 46) (74, 32) (26, 34) (25, 43) (44, 43) (29, 47) (50, 48) (58, 65) (53, 37) (43, 54) (68, 55) (37, 59) (52, 68) (43, 43) (56, 64) (47, 35) (59, 66) (62, 70) (34, 63) (38, 39) (60, 47) (73, 34) (25, 36) (69, 41) (68, 63) (73, 38) (44, 66) (26, 37) (27, 50) (46, 25) (32, 60) (40, 44) (52, 38) (32, 39) (28, 60) (41, 64) (57, 53) (69, 26) (37, 69) (59, 26) (32, 32) (45, 45) (32, 25)
	50	(51, 63) (53, 36) (68, 28) (49, 28) (43, 58) (35, 65) (51, 45) (26, 74) (33, 27) (27, 30) (65, 72) (30, 35) (63, 74) (49, 26) (54, 65) (71, 50) (58, 74) (63, 60) (69, 74) (61, 64) (25, 43) (40, 28) (64, 38) (67, 70) (27, 69) (71, 49) (59, 43) (46, 62) (59, 42) (56, 50) (73, 30) (74, 26) (55, 42) (50, 57) (65, 56) (72, 49) (49, 74) (36, 34) (46, 59) (62, 26) (55, 65) (48, 33) (56, 28) (73, 64) (68, 46) (25, 36) (53, 69) (27, 48) (71, 68) (67, 61)
	50	(74, 35) (32, 58) (66, 34) (25, 65) (53, 48) (46, 32) (64, 68) (29, 54) (64, 35) (48, 45) (50, 60) (64, 40) (26, 55) (39, 27) (48, 35) (29, 57) (67, 42) (40, 48) (70, 61) (45, 29) (65, 37) (26, 42) (69, 68) (46, 37) (46, 43) (57, 46) (56, 62) (65, 47) (51, 55) (40, 37) (45, 61) (57, 33) (73, 26) (41, 69) (48, 50) (68, 58) (45, 54) (32, 73) (48, 32) (40, 55) (65, 49) (31, 48) (32, 41) (58, 66) (67, 32) (33, 62) (38, 42) (37, 26) (70, 59) (29, 58)

B.2 Data Ukuran Besar

Ukuran Plat (P, L) = (200, 200); Ukuran Pola (p, l)

Macam Pola = m (antara 10 sampai 50)

Plat (P, L)	m	Pola (p, l)
(200, 200)	10	(148, 71) (65, 116) (132, 68) (51, 131) (106, 96) (93, 64) (129, 137) (59, 109) (129, 71) (97, 90)
	10	(100, 120) (128, 81) (53, 110) (79, 55) (96, 71) (58, 115) (135, 84) (80, 97) (141, 122) (91, 58)
	10	(130, 75) (53, 84) (138, 136) (93, 75) (92, 86) (115, 92) (113, 125) (131, 95) (103, 111) (80, 74)
	10	(90, 122) (115, 67) (146, 53) (82, 139) (96, 101) (137, 117) (91, 108) (64, 147) (97, 65) (81, 111)
	10	(131, 98) (63, 96) (64, 83) (116, 132) (135, 65) (67, 124) (77, 84) (74, 53) (140, 118) (58, 116)

Plat (P, L)	m	Pola (p, l)
(200, 200)	20	(51, 111) (108, 127) (69, 139) (131, 114) (98, 128) (124, 69) (87, 140) (102, 137) (122, 79) (109, 60) (84, 85) (92, 107) (141, 117) (123, 149) (85, 59) (90, 77) (142, 73) (88, 58) (102, 101) (103, 72)
	20	(69, 91) (146, 100) (107, 87) (87, 128) (141, 118) (107, 102) (142, 72) (52, 147) (99, 147) (70, 132) (59, 93) (56, 149) (88, 120) (136, 132) (89, 115) (77, 136) (68, 86) (91, 58) (146, 50) (68, 60)
	20	(56, 113) (77, 130) (61, 119) (90, 98) (102, 60) (143, 82) (145, 77) (129, 135) (92, 100) (78, 122) (83, 92) (93, 147) (86, 108) (120, 98) (67, 82) (91, 98) (85, 96) (56, 68) (149, 105) (95, 139)
	20	(132, 109) (91, 79) (134, 107) (78, 114) (138, 75) (112, 54) (97, 132) (136, 72) (147, 116) (120, 76) (140, 73) (97, 85) (130, 110) (52, 91) (72, 99) (101, 75) (105, 110) (83, 107) (126, 120) (66, 142)
	20	(94, 121) (105, 104) (101, 123) (139, 104) (110, 134) (92, 52) (144, 53) (123, 140) (75, 114) (143, 128) (55, 59) (85, 138) (70, 144) (108, 148) (129, 88) (52, 52) (110, 128) (96, 100) (123, 143) (77, 66)

Plat (P, L)	M	Pola (p, l)
(200, 200)	30	(63, 149)(63, 83)(65, 91)(99, 60)(88, 79)(77, 56)(79, 134)(74, 87) (108, 55)(76, 89)(57, 126)(91, 53)(121, 101)(94, 122)(50, 59) (74, 102)(130, 112)(76, 133)(97, 95)(54, 116)(116, 92)(73, 107) (85, 59)(135, 108)(133, 52)(116, 116)(134, 86)(132, 130)(141, 57) (76, 101)
	30	(124, 127)(95, 88)(66, 139)(97, 64)(142, 112)(109, 58)(94, 53) (119, 63)(69, 59)(84, 112)(93, 107)(80, 119)(91, 59)(121, 146) (71, 130)(92, 120)(148, 92)(53, 123)(60, 109)(111, 139)(110, 127) (81, 112)(137, 77)(76, 57)(87, 51)(82, 138)(67, 57)(137, 87) (65, 75)(97, 141)
	30	(92, 65)(62, 113)(80, 89)(107, 135)(133, 66)(124, 126)(119, 66) (85, 79)(142, 91)(71, 74)(53, 136)(133, 51)(68, 102)(78, 112) (104, 106)(96, 128)(67, 122)(74, 66)(138, 139)(59, 75)(139, 119) (62, 91)(93, 93)(128, 138)(91, 139)(106, 51)(138, 108)(89, 142) (141, 107)(114, 88)
	30	(62, 58)(149, 101)(146, 77)(90, 117)(96, 56)(58, 125)(64, 137) (71, 105)(65, 111)(113, 145)(147, 76)(51, 76)(69, 148)(145, 131) (64, 102)(78, 110)(100, 81)(82, 127)(58, 118)(139, 54)(89, 140) (98, 91)(129, 59)(117, 136)(78, 62)(140, 131)(58, 74)(145, 89) (135, 127)(144, 60)
	30	(50, 78)(121, 73)(90, 60)(117, 53)(116, 147)(121, 119)(105, 83) (117, 105)(86, 82)(89, 59)(79, 129)(103, 104)(52, 117)(100, 81) (79, 106)(71, 102)(140, 93)(114, 77)(107, 122)(71, 99)(115, 82) (109, 103)(93, 63)(101, 128)(120, 72)(93, 146)(125, 130)(91, 50) (143, 104)(96, 96)

Plat (P, L)	M	Pola (p, l)
(200, 200)	40	(104, 132)(59, 55)(113, 67)(50, 120)(138, 131)(55, 64) (135, 52)(143, 50)(60, 148)(78, 145)(50, 56)(56, 74)(80, 127) (89, 130)(124, 147)(86, 128)(70, 145)(86, 142)(77, 87) (143, 104)(76, 92)(77, 122)(137, 103)(135, 56)(70, 97)(125, 69) (125, 126)(127, 118)(117, 116)(77, 83)(74, 74)(90, 122)(94, 70) (73, 115)(76, 100)(125, 131)(123, 147)(140, 125)(66, 94) (66, 67)(143, 56)(61, 139)
	40	(53, 125)(102, 78)(53, 124)(90, 52)(83, 120)(149, 106)(86, 79) (143, 101)(148, 97)(88, 82)(61, 60)(52, 121)(83, 114)(84, 73) (64, 63)(120, 139)(105, 55)(82, 143)(80, 113)(149, 56)(108, 125) (89, 135)(84, 109)(77, 54)(93, 100)(126, 51)(61, 135)(109, 108) (131, 136)(68, 145)(57, 114)(113, 102)(57, 134)(65, 119)

	40	(66, 130)(108, 58)(90, 55)(139, 125)(140, 97)(133, 66)(93, 103) (70, 75)(137, 93)(88, 118)(57, 54)(54, 58)(99, 143)(69, 55) (117, 144)(52, 58)(61, 58)(73, 116)(104, 116)(114, 84)(86, 143) (94, 94)(126, 123)(84, 108)(143, 139)(91, 97)(76, 79)(91, 95) (101, 143)(145, 119)(77, 83)(118, 63)(71, 71)(92, 113)(56, 78) (148, 125)(64, 84)(139, 72)(148, 87)(102, 53)
	40	(89, 62)(65, 117)(136, 89)(50, 144)(82, 147)(93, 91)(112, 61) (105, 91)(62, 130)(59, 59)(84, 73)(95, 58)(120, 98)(144, 110) (101, 139)(127, 68)(145, 146)(101, 76)(117, 123)(121, 93) (69, 129)(62, 146)(80, 105)(127, 148)(55, 84)(115, 91)(62, 100) (132, 97)(111, 102)(77, 97)(110, 137)(134, 60)(136, 59)(138, 101) (133, 86)(113, 130)(96, 98)(92, 134)(121, 52)(70, 93)
	40	(132, 74)(145, 73)(93, 120)(53, 58)(101, 52)(105, 96)(79, 51) (67, 89)(148, 61)(59, 142)(129, 147)(58, 147)(106, 104)(123, 82) (58, 58)(87, 136)(130, 128)(79, 96)(116, 58)(75, 57)(60, 141) (88, 147)(101, 68)(140, 65)(122, 100)(91, 77)(87, 138)(107, 136) (50, 138)(88, 94)(97, 89)(103, 139)(145, 63)(93, 74)(101, 105) (133, 113)(68, 104)(81, 68)(146, 87)(138, 51)

Plat (P, L)	M	Pola (p, l)
(200, 200)	50	(58, 112)(108, 112)(115, 57)(130, 91)(85, 126)(114, 50)(127, 127) (113, 115)(132, 88)(108, 52)(51, 114)(85, 145)(113, 113)(137, 51) (69, 121)(118, 61)(115, 125)(129, 79)(61, 64)(120, 109)(78, 137) (126, 107)(119, 67)(83, 128)(102, 96)(91, 99)(102, 104)(107, 75) (88, 80)(101, 68)(110, 98)(103, 60)(106, 54)(118, 134)(101, 84) (88, 86)(102, 77)(143, 98)(129, 120)(149, 96)(124, 74)(54, 112) (123, 55)(96, 127)(97, 56)(115, 65)(89, 60)(91, 66)(107, 102) (84, 64)
	50	(145, 106)(68, 121)(138, 90)(80, 134)(93, 108)(117, 86)(127, 95) (52, 114)(116, 114)(69, 124)(149, 131)(117, 64)(108, 62) (148, 148)(148, 64)(82, 78)(136, 83)(110, 115)(77, 62)(146, 100) (131, 70)(62, 118)(94, 121)(90, 53)(116, 91)(65, 93)(64, 134) (139, 66)(56, 74)(110, 126)(117, 122)(112, 114)(51, 132)(89, 69) (116, 116)(147, 60)(147, 57)(133, 62)(68, 106)(115, 106)(89, 60) (140, 61)(68, 133)(109, 147)(101, 81)(84, 71)(63, 147)(119, 137) (98, 134)(97, 87)

	50	(59, 115)(82, 54)(125, 135)(145, 142)(52, 90)(118, 75)(136, 60) (75, 58)(94, 135)(50, 60)(95, 79)(147, 107)(89, 100)(95, 138) (118, 135)(148, 74)(119, 60)(148, 97)(134, 85)(86, 132)(102, 104) (140, 90)(89, 92)(67, 115)(84, 86)(80, 135)(74, 82)(147, 110) (58, 132)(74, 133)(87, 143)(76, 137)(117, 92)(136, 88)(127, 100) (115, 57)(136, 65)(64, 105)(115, 123)(135, 64)(88, 72)(112, 77) (54, 114)(101, 105)(91, 79)(139, 68)(111, 133)(135, 84)(81, 90) (131, 80)
	50	(136, 88)(58, 94)(127, 84)(117, 104)(112, 83)(118, 140)(63, 116) (57, 70)(137, 117)(98, 70)(64, 72)(116, 111)(129, 141)(99, 125) (148, 74)(117, 135)(78, 72)(114, 74)(108, 57)(76, 95)(112, 67) (77, 76)(74, 83)(118, 71)(105, 144)(96, 137)(115, 125)(77, 62) (116, 68)(63, 70)(62, 102)(69, 71)(54, 88)(52, 75)(68, 71)(59, 113) (78, 133)(91, 112)(94, 74)(95, 118)(125, 64)(124, 84)(137, 146) (138, 142)(77, 81)(142, 133)(75, 70)(60, 85)(147, 86)(110, 131)
	50	(54, 100)(130, 121)(108, 104)(60, 84)(104, 131)(83, 122)(146, 59) (81, 147)(81, 129)(64, 102)(149, 109)(72, 129)(146, 115)(58, 54) (120, 130)(98, 124)(74, 111)(102, 106)(131, 65)(106, 135)(93, 135) (72, 73)(100, 125)(136, 108)(56, 58)(61, 119)(104, 145)(109, 72) (142, 111)(125, 130)(123, 99)(95, 90)(88, 55)(128, 84)(110, 69) (68, 105)(57, 105)(148, 134)(101, 74)(128, 103)(131, 107)(82, 97) (147, 148)(127, 82)(140, 73)(120, 142)(87, 85)(52, 115)(119, 77) (110, 97)

LAMPIRAN C

CONTOH PENYELESAIAN SECARA MANUAL

Contoh :

Sebuah lembaran plate dengan ukuran panjang dan lebarnya adalah 9 x 11 akan dipotong untuk memenuhi order dengan pola-pola 3 x 3, 4 x 3 dan 2 x 4. Hitung profit untuk mendapatkan cutting pattern yang optimal?

Data:

Inisial plate :	$(P, L) = (9, 11)$
Himpunan pola (p_i, l_i) :	$S = \{(3, 3), (4, 3), (2, 4)\}$
Profit:	$\pi_i = p_i l_i$
Banyaknya pola:	$m = 3$

Penyelesaian:**Langkah 1 (menentukan titik-titik untuk pemotongan guillotine)**

$$P_p = \{x \mid x = \max \sum_{i=1}^m z_i p_i \leq [1/2 P], z_i \text{ integer positif}\},$$

$$P_l = \{y \mid y = \max \sum_{i=1}^m z_i p_i \leq [1/2 L], z_i \text{ integer positif}\},$$

untuk mendapatkan x dan y digenerate dengan pers. 1.

$$l = \{1, 2, 3\}, X = \{0, 1, 2, 3, 4\}$$

$$l = 1 \wedge x = 3;$$

$$\begin{aligned} \min_k \{g_{i-1}(x - k l)\} &= \min_k \{g_{1-1}(3 - 1 \times 3)\} \\ &= \min_k \{g_0(0)\} \\ &= 0 \end{aligned}$$

$$\max \{1, 0\} = 1$$

$$\begin{aligned} \min \{g_{i-1}(x), 1\} &= \min \{g_{1-1}(3), 1\} \\ &= \min \{g_0(3), 1\} \\ &= \min \{-, 1\} \\ &= 1 \end{aligned}$$

Jadi $x=3$ termasuk P_p .

$$l = 2 \wedge x = 4;$$

$$\begin{aligned} \min_k \{g_{i-1}(x - k l)\} &= \min_k \{g_{2-1}(4 - 1 \times 2)\} \\ &= \min_k \{g_1(2)\} \\ &= \min_k \{g_0(0)\} \\ &= 0 \end{aligned}$$

$$\max \{1, 0\} = 1$$

$$\begin{aligned} \min \{g_{i-1}(x), 1\} &= \min \{g_{2-1}(4), 1\} \\ &= \min \{g_1(4), 1\} \\ &= \min \{g_0(4), 1\} \\ &= \min \{-, 1\} \\ &= 1 \end{aligned}$$

Jadi $x=4$ termasuk P_p .

$$l = 3 \wedge x = 2;$$

$$\begin{aligned} \min_k \{g_{i-1}(x - k \cdot l)\} &= \min_k \{g_{3-1}(2 - 1 \times 2)\} \\ &= \min_k \{g_2(0)\} \\ &= \min_k \{g_2(0)\} \\ &= \min_k \{g_0(0)\} \\ &= 0 \end{aligned}$$

$$\text{maks} \{1, 0\} = 1$$

$$\begin{aligned} \min \{g_{i-1}(x), 1\} &= \min \{g_{3-1}(2), 1\} \\ &= \min \{g_2(2), 1\} \\ &= \min \{g_1(2), 1\} \\ &= \min \{g_0(2), 1\} \\ &= \min \{-, 1\} \\ &= 1 \end{aligned}$$

Jadi $x=2$ termasuk P_p .

Dari hasil generate untuk semua elemen l dan X , maka diperoleh $P_p = \{2, 3, 4\}$ dan $P_L = \{3, 4\}$

Langkah 2 (mengenerate strip horisontal tertinggi dan strip vertikal terlebar)

a. Strip horisontal tertinggi

$$S = \{(3, 3), (4, 3), (2, 4)\} \text{ (urut naik untuk lebar)}$$

$$S_{k1} = \{(3, 3)\}$$

$$S_{k2} = \{(3, 3), (4, 3)\}$$

$$S_{k3} = \{(3, 3), (4, 3), (2, 4)\}$$

b. Strip vertikal terlebar

$$S = \{(2, 4), (3, 3), (4, 3)\} \text{ (urut naik untuk panjang)}$$

$$S_{k1} = \{(2, 4)\}$$

$$S_{k2} = \{(2, 4), (3, 3)\}$$

$$S_{k3} = \{(2, 4), (3, 3), (4, 4)\}$$

$$n = 1$$

Langkah 3 (menentukan pemotongan guillotine)

Untuk pemotongan Guillotine vertikal di titik $X_j = 2$. Hasilnya membentuk 2 buah sub plate (2, 11) dan (7, 11).

Langkah 3. 1 (mengenerate strip-strip horisontal dan vertikal)

a. Strip-strip horisontal

Sub Plate (2, 11)

Syarat: $3X_1 \leq 2$ maka $X_1 = -$

$$\text{ODK (1)} = F_1(2) = \text{Maks} [(3 \times 3) X_1] = -$$

Syarat: $3X_1 + 4X_2 \leq 2$ maka $X_1 = X_2 = -$

$$\text{ODK (2)} = F_2(2) = \text{Maks} [(3 \times 3) X_1 + (4 \times 3) X_2] = -$$

Syarat: $3X_1 + 4X_2 + 2X_3 \leq 2$ maka $X_3 = 1$

$$\text{ODK (3)} = F_3(2) = \text{Maks} [(3 \times 3) X_1 + (4 \times 3) X_2 + (2 \times 4) X_3] = 2 \times 4 \times 1 = 8$$

Sub Plate (7, 11)

Syarat: $3X_1 \leq 7$ maka $X_1 = 2$

$$\text{ODK (1)} = F_1(7) = \text{Maks} [(3 \times 3) X_1] = 3 \times 3 \times 2 = 18$$

Syarat: $3X_1 + 4X_2 \leq 7$ maka $X_1 = 1$ dan $X_2 = 1$

$$\text{ODK (2)} = F_2(7) = \text{Maks} [(3 \times 3) X_1 + (4 \times 3) X_2] = (3 \times 3 \times 1) + (4 \times 3 \times 1) = 21$$

Syarat: $3X_1 + 4X_2 + 2X_3 \leq 7$ maka $X_1 = 1$ dan $X_3 = 2$

$$\text{ODK (3)} = F_3(7) = \text{Maks} [(3 \times 3) X_1 + (4 \times 3) X_2 + (2 \times 4) X_3] = (3 \times 3 \times 1) + (2 \times 4 \times 2) = 25$$

b. Strip-strip vertikal

Sub Plate (2, 11)

Syarat: $4X_1 \leq 11$ maka $X_1 = 2$

$$\text{ODK (1)} = F_1(11) = \text{Maks} [(2 \times 4) X_1] = 2 \times 4 \times 2 = 16$$

Syarat: $4X_1 + 3X_2 \leq 11$ maka $X_2 = 3$

$$\text{ODK (2)} = F_2(11) = \text{Maks} [(2 \times 4) X_1 + (3 \times 3) X_2] = 3 \times 3 \times 3 = 27$$

$$\text{Syarat: } 3X_1 + 4X_2 + 2X_3 = 11 \text{ maka } X_3 = 3$$

$$\text{ODK (3)} = F_3(11) = \text{Maks} [(2 \times 4) X_1 + (3 \times 3) X_2 + (4 \times 3) X_3] = 4 \times 3 \times 3 = 36$$

Sub Plate (7,11) sama dengan ODK (i) untuk sub plate (2,11)

Langkah 3.2 (Mengisi Plate secara horisontal dan vertikal)

a. Mengisi strip-strip horisontal

Sub plate (2,11)

$$\text{Syarat: } 3y_1 \leq 11 \text{ maka } y_1 = -$$

$$\text{ODK (4)} = F_4(11) = \text{Maks} (F_1(2) y_1) = -$$

$$\text{Syarat: } 3y_1 + 3y_2 \leq 11 \text{ maka } y_1 = y_2 = -$$

$$\text{ODK (5)} = F_5(11) = \text{Maks} (F_1(2) y_1 + F_2(2) y_2) = -$$

$$\text{Syarat: } 3y_1 + 3y_2 + 4y_3 \leq 11 \text{ maka } y_3 = 2$$

$$\text{ODK (6)} = F_6(11) = \text{Maks} (F_1(2) y_1 + F_2(2) y_2 + F_3(2) y_3) = 8 \times 2 = 16$$

Sub plate (7,11)

$$\text{Syarat: } 3y_1 \leq 11 \text{ maka } y_1 = 3$$

$$\text{ODK (4)} = F_4(11) = \text{Maks} (F_1(7) y_1) = 18 \times 3 = 54$$

$$\text{Syarat: } 3y_1 + 3y_2 \leq 11 \text{ maka } y_1 = 1 \text{ dan } y_2 = 2$$

$$\text{ODK (5)} = F_5(11) = \text{Maks} (F_1(7) y_1 + F_2(7) y_2) = 18 \times 1 + 21 \times 2 = 60$$

$$\text{Syarat: } 3y_1 + 3y_2 + 4y_3 \leq 11 \text{ maka } y_2 = 1 \text{ dan } y_3 = 2$$

$$\text{ODK (6)} = F_6(11) = \text{Maks} (F_1(7) y_1 + F_2(7) y_2 + F_3(7) y_3) = 21 \times 1 + 25 \times 2 = 71$$

b. Mengisi strip-strip vertikal

Sub plate (2,11)

$$\text{Syarat: } 2y_1 \leq 2 \text{ maka } y_1 = 1$$

$$\text{ODK (4)} = F_4(2) = \text{Maks} (F_1(11) y_1) = 16 \times 1 = 16$$

$$\text{Syarat: } 2y_1 + 3y_2 \leq 2 \text{ maka } y_1 = 1$$

$$\text{ODK (5)} = F_5(2) = \text{Maks} (F_1(11) y_1 + F_2(11) y_2) = 16$$

$$\text{Syarat: } 2y_1 + 3y_2 + 4y_3 \leq 2 \text{ maka } y_1 = 1$$

$$\text{ODK (6)} = F_6(2) = \text{Maks} (F_1(11) y_1 + F_2(11) y_2 + F_3(11) y_3) = 16$$

Sub plate (7,11)

$$\text{Syarat: } 2y_1 \leq 7 \text{ maka } y_1 = 3$$

$$\text{ODK (4)} = F_4(7) = \text{Maks} (F_1(11) y_1) = 16 \times 3 = 48$$

$$\text{Syarat: } 2y_1 + 3y_2 \leq 7 \text{ maka } y_1 = 2 \text{ dan } y_2 = 1$$

$$\text{ODK (5)} = F_5(7) = \text{Maks} (F_1(11) y_1 + F_2(11) y_2) = 16 \times 2 + 27 \times 1 = 59$$

$$\text{Syarat: } 2y_1 + 3y_2 + 4y_3 \leq 7 \text{ maka } y_2 = 1 \text{ dan } y_3 = 1$$

$$\text{ODK (6)} = F_6(7) = \text{Maks} (F_1(11) y_1 + F_2(11) y_2 + F_3(11) y_3) = 27 \times 1 + 36 \times 1 = 63$$

Langkah 3.3 (Mencari nilai profit untuk masing-masing sub plate yang maksimal)

Sub plate (2,11)

$$V_1 = \text{maks} (V_1^{\text{hor}}, V_1^{\text{ver}}) = \text{maks} (16, 16) = 16$$

(sub plate diisi strip-strip horisontal atau vertikal dengan pola (2 x 4) sebanyak 3 kali)

Sub plate (7,11)

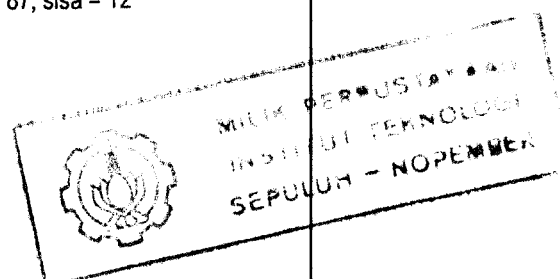
$$V_2 = \text{maks} (V_2^{\text{hor}}, V_2^{\text{ver}}) = \text{maks} (71, 63) = 71$$

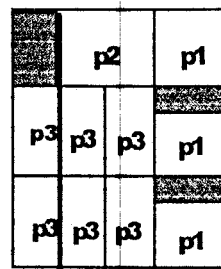
(sub plate diisi strip-strip horisontal dengan pola (3 x 3) dan (4x3) sebanyak 1 kali dan pola (3x3) dan (2,4) sebanyak 2 kali)

Langkah 3.4 (Mencari nilai profit plate yang maksimal)

$$V^3 = V_1 + V_2 = 16 + 71 = 87$$

Nilai optimal dari cutting pattern yang dihasilkan = 87, sisa = 12





$x_j = 2$

$n = 2$

Langkah 3 (menentukan pemotongan guillotine)

Untuk pemotongan Guillotine vertikal di titik $X_j = 3$. Hasilnya membentuk 2 buah sub plate (3, 11) dan (6, 11).

Langkah 3. 1 (mengenerate strip-strip horisontal dan vertikal)

a. Strip-strip horisontal

Sub Plate (3,11)

Syarat: $3X_1 \leq 3$, $X_1 = 1$

ODK (1) = $F_1(3) = \text{Maks} [(3 \times 3) X_1] = 3 \times 3 \times 1 = 9$

Syarat: $3X_1 + 4X_2 \leq 3$, $X_1 = 1$

ODK (2) = $F_2(3) = \text{Maks} [(3 \times 3) X_1 + (4 \times 3) X_2] = 3 \times 3 \times 1 = 9$

Syarat: $3X_1 + 4X_2 + 2X_3 \leq 3$, $X_1 = 1$

ODK (3) = $F_3(3) = \text{Maks} [(3 \times 3) X_1 + (4 \times 3) X_2 + (2 \times 4) X_3] = 3 \times 3 \times 1 = 9$

Sub Plate (6,11)

Syarat: $3X_1 \leq 6$, $X_1 = 2$

ODK (1) = $F_1(6) = \text{Maks} [(3 \times 3) X_1] = 3 \times 3 \times 2 = 18$

Syarat: $3X_1 + 4X_2 \leq 6$, $X_2 = 1$

ODK (2) = $F_2(6) = \text{Maks} [(3 \times 3) X_1 + (4 \times 3) X_2] = 4 \times 3 \times 1 = 12$

Syarat: $3X_1 + 4X_2 + 2X_3 \leq 6$, $X_3 = 3$

ODK (3) = $F_3(6) = \text{Maks} [(3 \times 3) X_1 + (4 \times 3) X_2 + (2 \times 4) X_3] = 2 \times 4 \times 3 = 24$

b. Strip-strip vertikal

Sub Plate (3,11)

Syarat: $4X_1 \leq 11$, $X_1 = 2$

ODK (1) = $F_1(11) = \text{Maks} [(2 \times 4) X_1] = 2 \times 4 \times 2 = 16$

Syarat: $4X_1 + 3X_2 \leq 11$, $X_2 = 3$

ODK (2) = $F_2(11) = \text{Maks} [(2 \times 4) X_1 + (3 \times 3) X_2] = 3 \times 3 \times 3 = 27$

Syarat: $3X_1 + 4X_2 + 2X_3 = 11$, $X_3 = 3$

ODK (3) = $F_3(11) = \text{Maks} [(2 \times 4) X_1 + (3 \times 3) X_2 + (4 \times 3) X_3] = 4 \times 3 \times 3 = 36$

Sub Plate (6,11) sama dengan ODK (i) untuk sub plate (3,11)

Langkah 3.2 (Mengisi Plate secara horisontal dan vertikal)

a. Mengisi strip-strip horisontal

Sub plate (3,11)

Syarat: $3y_1 \leq 11$, $y_1 = 3$

ODK (4) = $F_4(11) = \text{Maks} (F_1(3) y_1) = 9 \times 3 = 27$

Syarat: $3y_1 + 3y_2 \leq 11$, $y_1 = 3$

ODK (5) = $F_5(11) = \text{Maks} (F_1(3) y_1 + F_2(3) y_2) = 9 \times 3 = 27$

Syarat: $3y_1 + 3y_2 + 4y_3 \leq 11$, $y_1 = 1$ dan $y_3 = 2$

ODK (6) = $F_6(11) = \text{Maks} (F_1(3) y_1 + F_2(3) y_2 + F_3(3) y_3) = 9 \times 1 + 9 \times 2 = 27$

Sub plate (6,11)Syarat: $3y_1 \leq 11, y_1 = 3$

$$\text{ODK (4)} = F_4(11) = \text{Maks } (F_1(6) y_1) = 18 \times 3 = 54$$

Syarat: $3y_1 + 3y_2 \leq 11, y_1 = 2 \text{ dan } y_2 = 1$

$$\text{ODK (5)} = F_5(11) = \text{Maks } (F_1(6) y_1 + F_2(6) y_2) = 18 \times 2 + 12 \times 1 = 48$$

Syarat: $3y_1 + 3y_2 + 4y_3 \leq 11, y_1 = 1 \text{ dan } y_3 = 2$

$$\text{ODK (6)} = F_6(11) = \text{Maks } (F_1(6) y_1 + F_2(6) y_2 + F_3(6) y_3) = 18 \times 1 + 24 \times 2 = 66$$

b. Mengisi strip-strip vertikal**Sub plate (3,11)**Syarat: $2y_1 \leq 3, y_1 = 1$

$$\text{ODK (4)} = F_4(3) = \text{Maks } (F_1(11) y_1) = 16 \times 1 = 16$$

Syarat: $2y_1 + 3y_2 \leq 3, y_2 = 1$

$$\text{ODK (5)} = F_5(3) = \text{Maks } (F_1(11) y_1 + F_2(11) y_2) = 27 \times 1 = 27$$

Syarat: $2y_1 + 3y_2 + 4y_3 \leq 3, y_2 = 1$

$$\text{ODK (6)} = F_6(3) = \text{Maks } (F_1(11) y_1 + F_2(11) y_2 + F_3(11) y_3) = 27 \times 1 = 27$$

Sub plate (6,11)Syarat: $2y_1 \leq 6, y_1 = 3$

$$\text{ODK (4)} = F_4(6) = \text{Maks } (F_1(11) y_1) = 16 \times 3 = 48$$

Syarat: $2y_1 + 3y_2 \leq 6, y_2 = 2$

$$\text{ODK (5)} = F_5(6) = \text{Maks } (F_1(11) y_1 + F_2(11) y_2) = 27 \times 2 = 54$$

Syarat: $2y_1 + 3y_2 + 4y_3 \leq 6, y_2 = 2$

$$\text{ODK (6)} = F_6(6) = \text{Maks } (F_1(11) y_1 + F_2(11) y_2 + F_3(11) y_3) = 27 \times 2 = 54$$

Langkah 3.3 (Mencari nilai profit untuk masing-masing sub plate yang maksimal)**Sub plate (3,11)**

$$V_1 = \text{maks } (V_1^{\text{hor}}, V_1^{\text{ver}}) = \text{maks } (27, 27) = 27$$

(sub plate diisi strip-strip horisontal atau vertikal dengan pola (3 x 3) sebanyak 3 kali)

Sub plate (6,11)

$$V_2 = \text{maks } (V_2^{\text{hor}}, V_2^{\text{ver}}) = \text{maks } (66, 54) = 66$$

(sub plate diisi strip-strip horisontal dengan 2 buah pola (3 x 3) dan 3 buah pola (2,4) sebanyak 2 kali)

Langkah 3.4 (Mencari nilai profit plate yang maksimal)

$$V^3 = V_1 + V_2 = 27 + 66 = 93$$

Nilai optimal dari cutting pattern yang dihasilkan = 93, sisa = 6

p1	p3	p3	p3
p1	p3	p3	p3
p1	p1	p1	

$$x_j = 3$$

$n = 3$

Langkah 3 (menentukan pemotongan guillotine)

Untuk pemotongan Guillotine vertikal di titik $X_j = 4$. Hasilnya membentuk 2 buah sub plate (4, 11) dan (5, 11).

Langkah 3. 1 (mengenerate strip-strip horisontal dan vertikal)

a. Strip-strip horisontal

Sub Plate (4,11)

Syarat: $3X_1 \leq 4, X_1 = 1$

$$\text{ODK (1)} = F_1(4) = \text{Maks} [(3 \times 3) X_1] = 3 \times 3 \times 1 = 9$$

Syarat: $3X_1 + 4X_2 \leq 4, X_2 = 1$

$$\text{ODK (2)} = F_2(4) = \text{Maks} [(3 \times 3) X_1 + (4 \times 3) X_2] = 4 \times 3 \times 1 = 12$$

Syarat: $3X_1 + 4X_2 + 2X_3 \leq 4, X_3 = 3$

$$\text{ODK (3)} = F_3(4) = \text{Maks} [(3 \times 3) X_1 + (4 \times 3) X_2 + (2 \times 4) X_3] = 2 \times 4 \times 2 = 16$$

Sub Plate (5,11)

Syarat: $3X_1 \leq 5, X_1 = 1$

$$\text{ODK (1)} = F_1(5) = \text{Maks} [(3 \times 3) X_1] = 3 \times 3 \times 1 = 9$$

Syarat: $3X_1 + 4X_2 \leq 5, X_2 = 1$

$$\text{ODK (2)} = F_2(5) = \text{Maks} [(3 \times 3) X_1 + (4 \times 3) X_2] = 4 \times 3 \times 1 = 12$$

Syarat: $3X_1 + 4X_2 + 2X_3 \leq 5, X_1 = 1 \text{ dan } X_3 = 1$

$$\text{ODK (3)} = F_3(5) = \text{Maks} [(3 \times 3) X_1 + (4 \times 3) X_2 + (2 \times 4) X_3] = (3 \times 3 \times 1) + (2 \times 4 \times 1) = 17$$

b. Strip-strip vertikal

Sub Plate (4,11)

Syarat: $4X_1 \leq 11, X_1 = 2$

$$\text{ODK (1)} = F_1(11) = \text{Maks} [(2 \times 4) X_1] = 2 \times 4 \times 2 = 16$$

Syarat: $4X_1 + 3X_2 \leq 11, X_2 = 3$

$$\text{ODK (2)} = F_2(11) = \text{Maks} [(2 \times 4) X_1 + (3 \times 3) X_2] = 3 \times 3 \times 3 = 27$$

Syarat: $3X_1 + 4X_2 + 2X_3 \leq 11, X_3 = 3$

$$\text{ODK (3)} = F_3(11) = \text{Maks} [(2 \times 4) X_1 + (3 \times 3) X_2 + (4 \times 3) X_3] = 4 \times 3 \times 3 = 36$$

Sub Plate (5,11) sama dengan ODK (i) untuk sub plate (4,11)

Langkah 3.2 (Mengisi Plate secara horisontal dan vertikal)

a. Mengisi strip-strip horisontal

Sub plate (4,11)

Syarat: $3y_1 \leq 11, y_1 = 3$

$$\text{ODK (4)} = F_4(11) = \text{Maks} (F_1(4) y_1) = 9 \times 3 = 27$$

Syarat: $3y_1 + 3y_2 \leq 11, y_1 = 1 \text{ dan } y_2 = 2$

$$\text{ODK (5)} = F_5(11) = \text{Maks} (F_1(4) y_1 + F_2(4) y_2) = 9 \times 1 + 12 \times 2 = 33$$

Syarat: $3y_1 + 3y_2 + 4y_3 \leq 11, y_2 = 1 \text{ dan } y_3 = 2$

$$\text{ODK (6)} = F_6(11) = \text{Maks} (F_1(4) y_1 + F_2(4) y_2 + F_3(4) y_3) = 12 \times 1 + 17 \times 2 = 46$$

Sub plate (5,11)

Syarat: $3y_1 \leq 11, y_1 = 3$

$$\text{ODK (4)} = F_4(11) = \text{Maks} (F_1(5) y_1) = 8 \times 3 = 27$$

Syarat: $3y_1 + 3y_2 \leq 11, y_1 = 1 \text{ dan } y_2 = 2$

$$\text{ODK (5)} = F_5(11) = \text{Maks} (F_1(5) y_1 + F_2(5) y_2) = 9 \times 1 + 12 \times 2 = 33$$

Syarat: $3y_1 + 3y_2 + 4y_3 \leq 11, y_2 = 1 \text{ dan } y_3 = 2$

$$\text{ODK (6)} = F_6(11) = \text{Maks} (F_1(5) y_1 + F_2(5) y_2 + F_3(5) y_3) = 12 \times 1 + 17 \times 2 = 46$$

b. Mengisi strip-strip vertikal

Sub plate (4,11)

Syarat: $2y_1 \leq 4, y_1 = 2$

$$\text{ODK (4)} = F_4(3) = \text{Maks} (F_1(11) y_1) = 16 \times 2 = 32$$

Syarat: $2y_1 + 3y_2 \leq 4$, $y_1 = 2$

ODK (5) = $F_5(3) = \text{Maks } (F_1(11)y_1 + F_2(11)y_2) = 16 \times 2 = 32$

Syarat: $2y_1 + 3y_2 + 4y_3 \leq 3$, $y_3 = 1$

ODK (6) = $F_6(3) = \text{Maks } (F_1(11)y_1 + F_2(11)y_2 + F_3(11)y_3) = 36 \times 1 = 36$

Sub plate (5,11)

Syarat: $2y_1 \leq 5$, $y_1 = 2$

ODK (4) = $F_4(5) = \text{Maks } (F_1(11)y_1) = 16 \times 2 = 32$

Syarat: $2y_1 + 3y_2 \leq 5$, $y_1 = 1$ dan $y_2 = 1$

ODK (5) = $F_5(5) = \text{Maks } (F_1(11)y_1 + F_2(11)y_2) = 16 \times 1 + 27 \times 1 = 43$

Syarat: $2y_1 + 3y_2 + 4y_3 \leq 5$, $y_1 = 1$ dan $y_2 = 1$

ODK (6) = $F_6(5) = \text{Maks } (F_1(11)y_1 + F_2(11)y_2 + F_3(11)y_3) = 16 \times 1 + 27 \times 1 = 43$

Langkah 3.3 (Mencari nilai profit untuk masing-masing sub plate yang maksimal)

Sub plate (4,11)

$V_1 = \text{maks } (V_1^{\text{hor}}, V_1^{\text{ver}}) = \text{maks } (44, 36) = 44$

(sub plate diisi sebuah strip horizontal dengan pola (4x3) dan dua kali strip horizontal dengan 2 buah pola (2 x 4))

Sub plate (5,11)

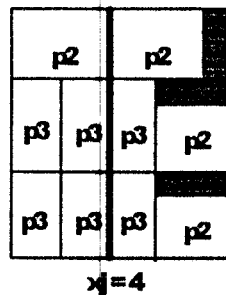
$V_2 = \text{maks } (V_2^{\text{hor}}, V_2^{\text{ver}}) = \text{maks } (46, 43) = 46$

(sub plate diisi sebuah strip horizontal dengan pola (4x3) dan dua kali strip horizontal dengan pola (3 x 3) dan (2 x 4))

Langkah 3.4 (Mencari nilai profit plate yang maksimal)

$V^4 = V_1 + V_2 = 44 + 46 = 90$

Nilai optimal dari cutting pattern yang dihasilkan = 90, sisa = 9



$n = 4$

Langkah 3 (menentukan pemotongan guillotine)

Untuk pemotongan Guillotine horizontal di titik $X_j = 3$. Hasilnya membentuk 2 buah sub plate (9, 3) dan (9,8).

Langkah 3.1 (mengenerate strip-strip horizontal dan vertikal)

a. Strip-strip horizontal

Sub Plate (9, 3)

Syarat: $3X_1 \leq 9$ maka $X_1 = 3$

ODK (1) = $F_1(9) = \text{Maks } [(3 \times 3)X_1] = 3 \times 3 \times 3 = 27$

Syarat: $3X_1 + 4X_2 \leq 9$ maka $X_1 = 3$

ODK (2) = $F_2(9) = \text{Maks } [(3 \times 3)X_1 + (4 \times 3)X_2] = 3 \times 3 \times 3 = 27$

Syarat: $3X_1 + 4X_2 + 2X_3 \leq 9$ maka $X_1 = 4$

ODK (3) = $F_3(9) = \text{Maks } [(3 \times 3)X_1 + (4 \times 3)X_2 + (2 \times 4)X_3] = 2 \times 4 \times 4 = 32$

Sub Plate (9,8) sama dengan ODK (i) untuk sub plate (9, 3)

b. Strip-strip vertikal**Sub Plate (9,3)**Syarat: $4X_1 \leq 3$ maka $X_1 = -$

$$\text{ODK (1)} = F_1(3) = \text{Maks} [(2 \times 4) X_1] = -$$

Syarat: $4X_1 + 3X_2 \leq 3$ maka $X_2 = 1$

$$\text{ODK (2)} = F_2(3) = \text{Maks} [(2 \times 4) X_1 + (3 \times 3) X_2] = 3 \times 3 \times 1 = 9$$

Syarat: $3X_1 + 4X_2 + 2X_3 = 3$ maka $X_3 = 1$

$$\text{ODK (3)} = F_3(3) = \text{Maks} [(2 \times 4) X_1 + (3 \times 3) X_2 + (4 \times 3) X_3] = (4 \times 3 \times 1) = 12$$

Sub Plate (9, 8)Syarat: $4X_1 \leq 8$ maka $X_1 = 2$

$$\text{ODK (1)} = F_1(8) = \text{Maks} [(2 \times 4) X_1] = 2 \times 4 \times 2 = 16$$

Syarat: $4X_1 + 3X_2 \leq 8$ maka $X_2 = 2$

$$\text{ODK (2)} = F_2(8) = \text{Maks} [(2 \times 4) X_1 + (3 \times 3) X_2] = 3 \times 3 \times 2 = 18$$

Syarat: $3X_1 + 4X_2 + 2X_3 = 8$ maka $X_3 = 2$

$$\text{ODK (3)} = F_3(8) = \text{Maks} [(2 \times 4) X_1 + (3 \times 3) X_2 + (4 \times 3) X_3] = (4 \times 3 \times 2) = 24$$

Langkah 3.2 (Mengisi Plate secara horisontal dan vertikal)**a. Mengisi strip-strip horisontal****Sub plate (9, 3)**Syarat: $3y_1 \leq 3$ maka $y_1 = 1$

$$\text{ODK (4)} = F_4(3) = \text{Maks} (F_1(9) y_1) = 27 \times 1 = 27$$

Syarat: $3y_1 + 3y_2 \leq 3$ maka $y_1 = 1$

$$\text{ODK (5)} = F_5(3) = \text{Maks} (F_1(9) y_1 + F_2(9) y_2) = 27 \times 1 = 27$$

Syarat: $3y_1 + 3y_2 + 4y_3 \leq 3$ maka $y_1 = 1$

$$\text{ODK (6)} = F_6(3) = \text{Maks} (F_1(9) y_1 + F_2(9) y_2 + F_3(9) y_3) = 27 \times 1 = 27$$

Sub plate (9,8)Syarat: $3y_1 \leq 8$ maka $y_1 = 2$

$$\text{ODK (4)} = F_4(8) = \text{Maks} (F_1(9) y_1) = 27 \times 2 = 54$$

Syarat: $3y_1 + 3y_2 \leq 11$ maka $y_1 = 2$

$$\text{ODK (5)} = F_5(8) = \text{Maks} (F_1(9) y_1 + F_2(9) y_2) = 54$$

Syarat: $3y_1 + 3y_2 + 4y_3 \leq 11$ maka $y_3 = 2$

$$\text{ODK (6)} = F_6(8) = \text{Maks} (F_1(9) y_1 + F_2(9) y_2 + F_3(9) y_3) = 32 \times 2 = 64$$

b. Mengisi strip-strip vertikal**Sub plate (9,3)**Syarat: $2y_1 \leq 9$ maka $y_1 = 4$

$$\text{ODK (4)} = F_4(9) = \text{Maks} (F_1(3) y_1) = -$$

Syarat: $2y_1 + 3y_2 \leq 9$ maka $y_2 = 3$

$$\text{ODK (5)} = F_5(9) = \text{Maks} (F_1(3) y_1 + F_2(3) y_2) = 9 \times 3 = 27$$

Syarat: $2y_1 + 3y_2 + 4y_3 \leq 9$ maka $y_2 = 3$

$$\text{ODK (6)} = F_6(9) = \text{Maks} (F_1(3) y_1 + F_2(3) y_2 + F_3(3) y_3) = 27$$

Sub plate (9,8)Syarat: $2y_1 \leq 9$ maka $y_1 = 4$

$$\text{ODK (4)} = F_4(9) = \text{Maks} (F_1(8) y_1) = 16 \times 4 = 64$$

Syarat: $2y_1 + 3y_2 \leq 9$ maka $y_1 = 3$ dan $y_2 = 1$

$$\text{ODK (5)} = F_5(9) = \text{Maks} (F_1(8) y_1 + F_2(8) y_2) = 16 \times 3 + 18 \times 1 = 66$$

Syarat: $2y_1 + 3y_2 + 4y_3 \leq 9$ maka $y_1 = y_2 = y_3 = 1$

$$\text{ODK (6)} = F_6(9) = \text{Maks} (F_1(8) y_1 + F_2(8) y_2 + F_3(8) y_3) = 16 \times 1 + 18 \times 1 + 24 \times 1 = 58$$

Langkah 3.3 (Mencari nilai profit untuk masing-masing sub plate yang maksimal)**Sub plate (9, 3)**

$$V_1 = \max(V_1^{\text{hor}}, V_1^{\text{ver}}) = \max(27, 27) = 27$$

(sub plate diisi strip-strip horisontal atau vertikal dengan 3 buah pola (3 x 3))

Sub plate (9, 8)

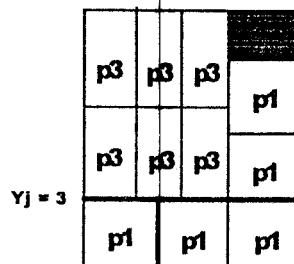
$$V_2 = \max(V_2^{\text{hor}}, V_2^{\text{ver}}) = \max(64, 66) = 66$$

(sub plate diisi strip-strip vertikal dengan dua buah pola (3 x 3) dan dua buah pola (2, 4) sebanyak 3 kali)

Langkah 3.4 (Mencari nilai profit plate yang maksimal)

$$V^3 = V_1 + V_2 = 27 + 66 = 93$$

Nilai optimal dari cutting pattern yang dihasilkan = 93, sisa = 6



$n = 5$

Langkah 3 (menentukan pemotongan guillotine)

Untuk pemotongan Guillotine horisontal di titik $X_j = 4$. Hasilnya membentuk 2 buah sub plate (9, 4) dan (9, 7).

Langkah 3. 1 (mengenerate strip-strip horisontal dan vertikal)**a. Strip-strip horisontal****Sub Plate (9, 4)**

Syarat: $3X_1 \leq 9$ maka $X_1 = 3$

$$\text{ODK (1)} = F_1(9) = \text{Maks} [(3 \times 3) X_1] = 3 \times 3 \times 3 = 27$$

Syarat: $3X_1 + 4X_2 \leq 9$ maka $X_1 = 3$

$$\text{ODK (2)} = F_2(9) = \text{Maks} [(3 \times 3) X_1 + (4 \times 3) X_2] = 3 \times 3 \times 3 = 27$$

Syarat: $3X_1 + 4X_2 + 2X_3 \leq 9$ maka $X_1 = 1$ dan $X_3 = 3$

$$\text{ODK (3)} = F_3(9) = \text{Maks} [(3 \times 3) X_1 + (4 \times 3) X_2 + (2 \times 4) X_3] = (3 \times 3 \times 1) + (2 \times 4 \times 3) = 33$$

Sub Plate (9, 7) sama dengan ODK (i) untuk sub plate (9, 4)

b. Strip-strip vertikal**Sub Plate (9, 4)**

Syarat: $4X_1 \leq 4$ maka $X_1 = 1$

$$\text{ODK (1)} = F_1(4) = \text{Maks} [(2 \times 4) X_1] = 2 \times 4 \times 1 = 8$$

Syarat: $4X_1 + 3X_2 \leq 4$ maka $X_2 = 1$

$$\text{ODK (2)} = F_2(4) = \text{Maks} [(2 \times 4) X_1 + (3 \times 3) X_2] = 3 \times 3 \times 1 = 9$$

Syarat: $3X_1 + 4X_2 + 2X_3 \leq 4$ maka $X_3 = 1$

$$\text{ODK (3)} = F_3(4) = \text{Maks} [(2 \times 4) X_1 + (3 \times 3) X_2 + (4 \times 3) X_3] = (4 \times 3 \times 1) = 12$$

Sub Plate (9, 7)

Syarat: $4X_1 \leq 7$ maka $X_1 = 1$

$$\text{ODK (1)} = F_1(7) = \text{Maks} [(2 \times 4) X_1] = 2 \times 4 \times 1 = 8$$

Syarat: $4X_1 + 3X_2 \leq 7$ maka $X_2 = 2$

$$\text{ODK (2)} = F_2(7) = \text{Maks} [(2 \times 4) X_1 + (3 \times 3) X_2] = 3 \times 3 \times 2 = 18$$

Syarat: $3X_1 + 4X_2 + 2X_3 \leq 7$ maka $X_3 = 2$

$$\text{ODK (3)} = F_3(7) = \text{Maks} [(2 \times 4) X_1 + (3 \times 3) X_2 + (4 \times 3) X_3] = (4 \times 3 \times 2) = 24$$

Langkah 3.2 (Mengisi Plate secara horisontal dan vertikal)

a. Mengisi strip-strip horisontal

Sub plate (9, 4)

Syarat: $3y_1 \leq 4$ maka $y_1 = 1$

$$\text{ODK (4)} = F_4(4) = \text{Maks} (F_1(9) y_1) = 27 \times 1 = 27$$

Syarat: $3y_1 + 3y_2 \leq 4$ maka $y_1 = 1$

$$\text{ODK (5)} = F_5(4) = \text{Maks} (F_1(9) y_1 + F_2(9) y_2) = 27 \times 1 = 27$$

Syarat: $3y_1 + 3y_2 + 4y_3 \leq 4$ maka $y_3 = 1$

$$\text{ODK (6)} = F_6(4) = \text{Maks} (F_1(9) y_1 + F_2(9) y_2 + F_3(9) y_3) = 33 \times 1 = 33$$

Sub plate (9, 7)

Syarat: $3y_1 \leq 7$ maka $y_1 = 2$

$$\text{ODK (4)} = F_4(7) = \text{Maks} (F_1(9) y_1) = 27 \times 2 = 54$$

Syarat: $3y_1 + 3y_2 \leq 7$ maka $y_1 = 2$

$$\text{ODK (5)} = F_5(7) = \text{Maks} (F_1(9) y_1 + F_2(9) y_2) = 54$$

Syarat: $3y_1 + 3y_2 + 4y_3 \leq 7$ maka $y_1 = 1$ dan $y_3 = 1$

$$\text{ODK (6)} = F_6(7) = \text{Maks} (F_1(9) y_1 + F_2(9) y_2 + F_3(9) y_3) = 27 \times 1 + 33 \times 1 = 60$$

b. Mengisi strip-strip vertikal

Sub plate (9, 4)

Syarat: $2y_1 \leq 9$ maka $y_1 = 4$

$$\text{ODK (4)} = F_4(9) = \text{Maks} (F_1(4) y_1) = 8 \times 4 = 32$$

Syarat: $2y_1 + 3y_2 \leq 9$ maka $y_1 = 3$ dan $y_2 = 1$

$$\text{ODK (5)} = F_5(9) = \text{Maks} (F_1(4) y_1 + F_2(4) y_2) = 8 \times 3 + 9 \times 1 = 33$$

Syarat: $2y_1 + 3y_2 + 4y_3 \leq 9$ maka $y_1 = y_2 = y_3 = 1$

$$\text{ODK (6)} = F_6(9) = \text{Maks} (F_1(4) y_1 + F_2(4) y_2 + F_3(4) y_3) = 8 \times 1 + 9 \times 1 + 12 \times 1 = 29$$

Sub plate (9, 7)

Syarat: $2y_1 \leq 9$ maka $y_1 = 4$

$$\text{ODK (4)} = F_4(9) = \text{Maks} (F_1(7) y_1) = 8 \times 4 = 32$$

Syarat: $2y_1 + 3y_2 \leq 9$ maka $y_2 = 3$

$$\text{ODK (5)} = F_5(9) = \text{Maks} (F_1(7) y_1 + F_2(7) y_2) = 18 \times 3 = 54$$

Syarat: $2y_1 + 3y_2 + 4y_3 \leq 9$ maka $y_2 = 3$

$$\text{ODK (6)} = F_6(9) = \text{Maks} (F_1(7) y_1 + F_2(7) y_2 + F_3(7) y_3) = 18 \times 3 = 54$$

Langkah 3.3 (Mencari nilai profit untuk masing-masing sub plate yang maksimal)

Sub plate (9, 4)

$$V_1 = \text{maks} (V_1^{\text{hor}}, V_1^{\text{ver}}) = \text{maks} (33, 33) = 33$$

(sub plate diisi strip-strip horisontal atau vertikal dengan 3 buah pola (2, 4) dan (3, 3))

Sub plate (9, 7)

$$V_2 = \text{maks} (V_2^{\text{hor}}, V_2^{\text{ver}}) = \text{maks} (60, 54) = 60$$

(sub plate diisi strip-strip horisontal dengan 3 buah pola (3, 3) dan 3 buah pola (2, 4) dan pola (3, 3))

Langkah 3.4 (Mencari nilai profit plate yang maksimal)

$$V^3 = V_1 + V_2 = 33 + 60 = 93$$

Nilai optimal dari cutting pattern yang dihasilkan = 92, sisa = 7

$y_j = 4$

p1	p1	p1	
p1	p3	p3	p3
p1	p3	p3	p3

Fase 0 - cut untuk Plate (9,11)

Langkah 3.1 (Mengenerate strip-strip horisontal dan vertikal)**a. Strip-strip Horisontal**

Syarat: $3X_1 \leq 9$ maka $X_1 = 3$

$$\text{ODK (1)} = F_1(9) = \text{Maks} [(3 \times 3) X_1] = 3 \times 3 \times 3 = 27$$

Syarat: $3X_1 + 4X_2 \leq 9$ maka $X_1 = 3$

$$\text{ODK (2)} = F_2(9) = \text{Maks} [(3 \times 3) X_1 + (4 \times 3) X_2] = 3 \times 3 \times 3 = 27$$

Syarat: $3X_1 + 4X_2 + 2X_3 \leq 9$ maka $X_1 = 1$ dan $X_3 = 3$

$$\text{ODK (3)} = F_3(9) = \text{Maks} [(3 \times 3) X_1 + (4 \times 3) X_2 + (2 \times 4) X_3] = (3 \times 3 \times 1) + (2 \times 4 \times 3) = 33$$

b. Strip-strip Vertikal

Syarat: $4X_1 \leq 11$, $X_1 = 2$

$$\text{ODK (1)} = F_1(11) = \text{Maks} [(2 \times 4) X_1] = 2 \times 4 \times 2 = 16$$

Syarat: $4X_1 + 3X_2 \leq 11$, $X_2 = 3$

$$\text{ODK (2)} = F_2(11) = \text{Maks} [(2 \times 4) X_1 + (3 \times 3) X_2] = 3 \times 3 \times 3 = 27$$

Syarat: $3X_1 + 4X_2 + 2X_3 \leq 11$, $X_3 = 3$

$$\text{ODK (3)} = F_3(11) = \text{Maks} [(2 \times 4) X_1 + (3 \times 3) X_2 + (4 \times 3) X_3] = 4 \times 3 \times 3 = 36$$

Langkah 3.2 (Mengisi strip-strip horisontal dan vertikal)**a. Horisontal**

Syarat: $3y_1 \leq 11$ maka $y_1 = 3$

$$\text{ODK (4)} = F_4(11) = \text{Maks} (F_1(9) y_1) = 27 \times 3 = 81$$

Syarat: $3y_1 + 3y_2 \leq 11$ maka $y_1 = 3$

$$\text{ODK (5)} = F_5(11) = \text{Maks} (F_1(9) y_1 + F_2(9) y_2) = 81$$

Syarat: $3y_1 + 3y_2 + 4y_3 \leq 11$ maka $y_1 = 1$ dan $y_3 = 2$

$$\text{ODK (6)} = F_6(11) = \text{Maks} (F_1(9) y_1 + F_2(9) y_2 + F_3(9) y_3) = 27 \times 1 + 33 \times 2 = 93$$

b. Vertikal

Syarat: $2y_1 \leq 9$ maka $y_1 = 4$

$$\text{ODK (4)} = F_4(9) = \text{Maks} (F_1(11) y_1) = 16 \times 4 = 64$$

Syarat: $2y_1 + 3y_2 \leq 9$ maka $y_2 = 3$

$$\text{ODK (5)} = F_5(9) = \text{Maks} (F_1(11) y_1 + F_2(11) y_2) = 27 \times 3 = 81$$

Syarat: $2y_1 + 3y_2 + 4y_3 \leq 9$ maka $y_1 = y_2 = y_3 = 1$

$$\text{ODK (6)} = F_6(9) = \text{Maks} (F_1(11) y_1 + F_2(11) y_2 + F_3(11) y_3) = 16 \times 1 + 27 \times 1 + 36 \times 1 = 79$$

Langkah 3.3 (Mencari nilai profit untuk masing-masing sub plate yang maksimal)

$$V_2 = \text{maks} (V_2^{\text{hor}}, V_2^{\text{ver}}) = \text{maks} (93, 81) = 93$$

(sub plate diisi strip horisontal dengan 3 buah pola (3 x 3) sekali dan strip horisontal dengan 3 buah pola (2,4) dan satu buah pola (3, 3) sebanyak 2 kali)

Langkah 3.4 (Mencari nilai profit plate yang maksimal)

$$V^0 = V_2^{\text{hor}} = 93$$

Nilai optimal dari cutting pattern yang dihasilkan = 93, sisa = 6

Langkah 4. (Menyimpan nilai cutting pattern yang optimal)

$X_j = 3$ (pemotongan guillotine vertikal)

$V = 93$ sisa 6

p1		p1	p1
p3	p3	p3	p1
p3	p3	p3	p1